Generic Topology Optimization Based on Local State Features

Dem Fachbereich
Elektrotechnik und Informationstechnik
der Technischen Universität Darmstadt
zur Erlangung des akademischen Grades
eines Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Dissertation

von

Dipl.-Ing. Nikola Aulig

geboren am 6. Februar 1984 in Karlsruhe

Referent:	Prof. Dr.-Ing. J. Adamy
Korreferent:	Prof. Dr. rer. nat. B. Sendhoff
Tag der Einreichung:	30. November 2016
Tag der mündlichen Prüfung:	4. Mai 2017

D17
Darmstadt 2017

Acknowledgements

The presented research was conducted at the Honda Research Institute Europe GmbH (HRI-EU). The institute provided a unique and excellent personal, scientific and technical environment that made this particular thesis possible.

First of all, I thank my supervisor Markus Olhofer for his ideas, scientific and formal support and continuous, optimistic attitude over all the years. I thank the HRI-EU president Prof. Bernhard Sendhoff for important advice and guidance, despite of his full calender and the formal, long-term commitment to this project. Special thanks for guidance and encouragement go to Prof. Jürgen Adamy from the Control Methods and Robotics (RMR) department, who offered to supervise the dissertation in a collaboration between the Technische Universität Darmstadt and HRI-EU. Most of all, I am thankful for the trust, which these three scientists put in me.

I would like to express my thanks to many colleagues and friends for their valuable support: Mariusz Bujny, Duane Detwiler, Fabian Duddeck, Giles Endicott, Michael Gienger, Lars Gräning, Thomas Guthier, Martina Hasenjäger, Matthew Kent, Ingolf Lepenies, Stefan Menzel, Emily Nutwell, Matthias Platho, Edgar Reehuis, Tobias Rodemann, Andrea Schnall, Olga Smalikho, Till Steiner, Thomas Weisswange, and all my colleagues from the Complex System Optimization and Analysis Group. Thank you for your - scientific and non-scientific - comments, ideas, discussions, encouragements, distractions, collaborations and too many other things to list them all. Some of these, as well as Damien Schlarb, Svenja Schlarb, Friedrich Schenk helped to improve the language throughout this document. Furthermore, I would like to thank the administrative and technical staff at HRI-EU and RMR for their support, especially Birgit Heid for organizing formalities and Burkhard Zittel for maintaining the computational cluster. I thank Prof. Andrew Ng for his excellent online lecture on machine learning.

Finally, I am very grateful to all my other friends, my parents and family, and especially Andrea Schnall for their invaluable support, comprehension and, most of all, just for being there all these years.

Contents

List of Figures

List of Tables

Symbols and Abbreviations

Symbols

ρ	Density, topology optimization design variable
F	Objective function
$\mathbf{u}$	State vector
G	Constraint
Ω	Design domain
$\mathbf{x}$	Point in design domain
V	Volume of structure
L	Number of extra constraints
ρ	Vector of densities, topology optimization design variables
N	Number of finite elements in design space mesh
E	Material Young's modulus
p	Penalization coefficient
c	Compliance
$\mathbf{l}$	Load vector
$\mathbf{K}$	Stiffness matrix
f	Volume fraction
v	Finite element volume
μ	Number of parent individuals
ϱ	Number of parent individuals used for creating new offspring individuals
κ	Individual lifetime
λ	Number of offspring individuals
$\boldsymbol{\theta}$	Vector of ES design variables, update-signal model parameters
k	Iteration number
Θ	Number of search dimensions in ES
$\mathbf{z}$	Random mutation vector
$\mathbf{C}$	Covariance matrix of ES mutation operator
$\mathcal{N}$	Normal distribution
σ	Standard deviation of normal distribution, global step size
$\boldsymbol{\sigma}$	Vector of strategy parameters
z	Random mutation number

τ	Learning parameter
$\mathbf{m}$	Distribution mean
$\mathbf{v}$	Adjoint state vector
S	Update-signal model
J	Number of LSFs
$\mathbf{s}_i$	LSF Vector
u	Nodal displacement
B	OC-input
m	Move limit
η	OC damping parameter
$\widehat{S}$	Filtered update-signal model
Λ	Lagrange multiplier for volume constraint
$\widehat{H}$	Sensitivity filter weights
$r_{\min}$	Sensitivity filter radius
ΔF	Design improvement
M	Number of evluations
g	MLP activation function
H	Number of hidden neurons
$\mathcal{V}$	Data set of LSF samples
$\mathbf{c}$	LSF prototype
P	Number of LSF prototypes
Ψ	Mapping from LSF space to finite set of indices
ζ	Cluster index
ϵ	Finite difference step length
T	Number of training samples
y	Prediction target
$\mathcal{G}$	Set of element or node ids
$\hat{k}$	SVR kernel function
$\hat{L}$	Number of support vectors
ν	Poisson's ratio
SED	Strain energy density
ϱ'	Volumetric mass density
σ_{Y}	Yield stress
E_{h}	Elasticity hardening modulus
q	Penalization coefficient for plasticity
$\mathrm{E}_{\mathrm{abs}}$	Energy absorption
IED	Internal energy density
$\mathbf{r}$	Simulation residual
t	Time

I Intrusion

Abbreviations

BESO	Bi-directional Evolutionary Structural Optimization
CMA	Covariance Matrix Adaptation
CPPN	Compositional Pattern Producing Network
EA	Evolutionary Algorithm
EC	Evolutionary Computation
ES	Evolution Strategy
ESL	Equivalent Static Load
FD	Finite Difference
FEA	Finite Element Analysis
HCA	Hybrid Cellular Automata
LIN	Linear Regression
LSF	Local State Feature
LSM	Level Set Method
MLP	Multi-Layer Perceptron
NE	Neuro-Evolution
NEAT	Neuro Evolution for Augmenting Topologies
OC	Optimality Criteria
PCM	Piecewise-Constant Model
RBF	Radial Basis Function
SERA	Sequential Element Rejections and Admissions
SVR	Support Vector Regression
SIMP	Solid Isotropic Material with Penalization
TOPAS	Topology Optimization by Predicting Aggregated Sensitivities
TOPS	Topology Optimization by Predicting Sensitivities

Abstract

The automatic creation of optimal concepts for mechanical structures in the computer-aided design process has become an important area of research. Continuum topology optimization methods determine the distribution of material within a pre-defined design space and, thus, not only the shape, but also the fundamental geometric layout of a structure. For this task, the majority of the existing, numerical optimization methods requires mathematical gradient information. However, when addressing optimization problems that involve highly non-linear or black-box simulations, it can be difficult to obtain satisfactory results or gradient information at all. In order to provide design concepts also for these types of problems, this thesis presents a generic topology optimization approach. The novel method realizes a self-contained learning component that utilizes physical simulation data to generate a search direction. Based on a continuous problem formulation, every design variable is improved iteratively by a learned update-signal. The individual update-signals are computed from local state features and substitute sensitivities of the design variables. Evolutionary optimization or supervised learning adapt the model parameters for determination of the update-signals to the chosen optimization goal. In empirical studies, the novel method reproduces reference structures with minimum compliance. When applied to a practical problem from the challenging domain of vehicle crashworthiness optimization, specifically the minimization of intrusion, it provides superior design concepts when compared to a frequently applied heuristic method. The results confirm that the proposed method is capable to yield innovative solutions to so far unsolved topology optimization problems.

Zusammenfassung

Die automatische Erstellung von optimalen Entwurfskonzepten für mechanische Strukturen im rechnergestützten Entwicklungsprozess ist ein wichtiger Forschungszweig. Methoden der Topologieoptimierung bestimmen die Materialverteilung in einem vordefinierten Entwurfsraum und daher nicht nur die Form, sondern auch die grundsätzliche geometrische Ausgestaltung einer Struktur. Die Mehrheit der verfügbaren numerischen Optimierungsmethoden benötigen hierfür mathematische Gradienteninformation. Betrachtet man jedoch Optimierungsprobleme, die stark nichtlineare oder Blackbox-Simulationen beinhalten, kann es schwierig sein, zufriedenstellende Ergebnisse oder überhaupt Gradienteninformation zu erhalten. Um auch für solche Probleme Entwurfskonzepte zu finden, wird in dieser Dissertation ein generischer Topologieoptimierungsansatz präsentiert. Die neue Methode realisiert eine eigenständige Lernkomponente, welche in der Lage ist, aus physikalischen Simulationsdaten eine Suchrichtung zu erstellen. Basierend auf einer kontinuierlichen Formulierung des Problems wird jede Entwurfsvariable durch ein gelerntes Updatesignal iterativ verbessert. Die individuellen Updatesignale berechnen sich aus lokalen Zustandsmerkmalen und ersetzen die Sensitivitäten der Entwurfsvariablen. Evolutionäre Optimierung oder überwachte Lernverfahren passen die Modellparameter zur Bestimmung der Updatesignale an das gewählte Optimierungsziel an. In empirischen Studien reproduziert die neue Methode Referenzstrukturen mit minimaler Nachgiebigkeit. Bei der Anwendung auf ein Problem aus dem anspruchsvollen Gebiet der Optimierung des Fahrzeug-Unfallverhaltens, speziell der Minimierung der Eindringtiefe, liefert sie überlegene Entwurfsvorschläge im Vergleich mit einer häufig verwendeten heuristischen Methode. Die Ergebnisse bestätigen, dass die vorgeschlagene Methode in der Lage ist, innovative Lösungen für bisher ungelöste Topologieoptimierungsprobleme zu erzeugen.

1 Introduction

Human engineers or designers usually judge their designs with respect to numerous practical and sometimes subjective criteria with the goal of finding a high quality solution that considers all necessary restrictions. They typically change design parameters in an iterative and intuitive process. In contrast to this, methods of design optimization aim at finding the best design parameters in order to maximize or minimize an objective performance or cost function, typically subject to a number of constraints.

Modern product-development processes use computational design and engineering tools for digital conceptualization and simulation. The economical advantages include a reduction of the number of costly prototypes and an overall acceleration of the development processes. Tools, such as finite element analysis for the prediction of structural performance measures have made computer-aided engineering methods a standard in the automotive and other industries. Increasing computing power results in the ability to perform more accurate simulations of complicated physical systems. This, in combination with competitive pressure, leads to an increased demand for systematic design optimization that replaces experience-based engineering practices.

Many engineering problems can be addressed as optimization tasks, for example tuning vehicle shape parameters to minimize the aerodynamic drag, or the thickness of metal sheets in the vehicle body to reduce weight. The advantage of design optimization methods over more intuitive strategies is that they improve solutions for predefined objectives and constraints in a formalized way. Since optimal solutions can rarely be found analytically, optimization requires the choice of an appropriate numerical algorithm for the iterative search of an optimum. Its computerised implementation facilitates to tune many design parameters automatically, even without detailed knowledge about the considered problem.

Evolutionary algorithms from the domain of computational intelligence are useful, global search methods that are frequently applied to practical design optimization tasks. They mimic biological evolution and improve solutions by random variations that are selected whenever they are favourable. This stochastic component enables the application of evolu-

tionary algorithms to a large variety of problem classes. Their universality is of particular interest for complex problems with strongly non-linear characteristics for which a specialized optimization approach is unavailable or unable to provide satisfactory results. Also problems that involve black-box simulations, in which only input and output are known but not the details of the analysis, can be addressed, since no assumptions on the problem are required.

Optimization methods have an especially high potential for quality enhancement when applied early in a product design process. *Topology Optimization* addresses the task of finding a conceptual design for a mechanical structure. The subject of such optimization is the basic layout and geometric connectivity of the structure, i.e. to find the optimum distribution of material and void domains within a predefined design space. Naturally, this is one of the first steps in the design process and provides remarkable potential for finding innovative and efficient concepts in an automatized fashion. The last decade has seen a tremendous rise in the number of publications in the field of topology optimization. This trend is reflected and enforced by an increasing application of topology optimization algorithms within industries that target lightweight designs, for instance civil, automotive or aerospace industries.

Conventional topology optimization methods rely on a deep theoretical understanding of the considered problem and a corresponding mathematical formulation. This knowledge makes it possible to devise algorithms that iteratively update the design parameters towards improvement. The update is based on the mathematical sensitivities of the optimization target with respect to the parameters. Hence, conventional topology optimization algorithms are so-called gradient-based algorithms. When using mathematical gradients, it is computationally cheap to determine a search direction towards a local optimum, resulting in efficient, yet specialized optimization procedures. However, practical design optimization problems are often highly complex and/or rely on black-box simulations. Consequently, explicit mathematical formulations are not available, and alternative, more comprehensive optimization methods are required. As mentioned above, evolutionary algorithms are suitable for such problems. However, the general applicability of evolutionary algorithms comes with relatively large computational cost, even when the number of parameters is reduced significantly. In the case of topology optimization, the inherently very high number of design parameters limits the direct applicability of evolutionary algorithms.

Therefore, this thesis explores a novel, generic concept for topology op-

timization. The project consists in using methods of computational intelligence for creating an algorithm that devises its own search direction independently from pre-defined mathematical gradients. In the words of Robert Le Ricolais[1], "*The art of structure is where to put the holes*". In conventional topology optimization algorithms, the knowledge causing the formation of voids and material domains is contained in the mathematical formulation. This implies that the crucial information is determined *prior* to the optimization. Any intelligent, generic topology optimization method has to autonomously discover or at least adequately substitute this knowledge.

Intuitive understanding of this task and inspiration for the aspired research direction is gained by the observation of biological processes, such as the growth of trees or the remodelling process in human bones. These biological processes have lead to nature-inspired topology optimization methods that are based on simple rules; for instance weakening of material in locations with low stress and stiffening of material in locations with high stress. Such basic rules result in efficient structural designs and have emerged in biological evolution without previous knowledge of the task and environment. Typically, the nature-inspired methods use local, structural state information, an aspect that is also inherent to rigorously mathematical methods. This leads to the present project's goal of establishing generic criteria for performing topology optimization.

Following the biological inspiration, a computational model of evolution or learning should be able to devise heuristic update-signals that indicate how to improve the design parameters. The proposed model is required to - in a systematic way - harness structural state information, which is obtained from the simulation. Essentially, such a method eliminates the necessity of a predefined mathematical model that is limited to only a single problem. This approach raises a number of methodological questions, most pressing of all how the update-signal is to be devised: Which mathematical models should be used? How should the model parameters be modified so that the model can accommodate the aforementioned, pertinent information? And what should be the physical input features?

Computational concepts for nature-inspired problem-solving methods are candidates for addressing these questions. They can be found in the domain of Computational Intelligence, for instance, evolutionary algorithms for optimization, or artificial neural networks for learning tasks. Learning methods, such as artificial neural network models are capable to perform

[1] French-American engineer and philosopher, 1894-1977

difficult prediction and pattern recognition tasks. The applications include identifying and classifying objects in images for a vast number of problems, decision-making tasks for instance in games, as well as contributions to autonomous driving. In the considered context of topology optimization, artificial neural networks and statistical models will be utilized to process the structural state information and provide the required update-signals. Instead of addressing the topology optimization problem directly, evolutionary algorithms are very adequate for optimization parameters of these update strategies.

Figure 1.1 illustrates the different topology optimization approaches and the research focus of this thesis. The central contribution of this thesis is the proposal of a novel methodology for generic topology optimization.

Figure 1.2 shows an overview of the structure of the thesis. Subsequent to this introduction, the work is divided into the following chapters:

Chapter 2 provides the fundamentals on topology optimization and evolutionary computation that are scientific prerequisites for the proposed, novel methods. The description of topology optimization approaches focuses a the general, density-based formulation that provides a foundation for the proposed generic method. Furthermore, evolutionary computation methods are relevant for the introduced learning component. The section emphasizes evolution strategies, which are most suited for the targeted real-valued parameter optimizations.

Chapter 3 continues with an overview and categorization of approaches for representing structures for topology optimization with evolutionary computation. For the application of evolutionary algorithms, an essential issue is how to represent the structure, since the computational cost increases with the number of design parameters. The chapter introduces three different categories: grid, geometric and indirect representations. These categories are used to comprehensively classify the approaches from literature. Furthermore, the breakdown enables to outline assets and limitations of the different representation types. The chapter closes with a discussion on the relation to gradient-based algorithms and existing as well as potential applications.

Chapter 4 is the backbone of this thesis, since it introduces the methodological novelties for generic topology optimization. The initial, general explanation introduces a model that determines update-signals based on simulation data. The proposed algorithm performs topology optimization and, each iteration, improves the design parameters of the structure according to the model outputs. Subsequently, the chapter presents two

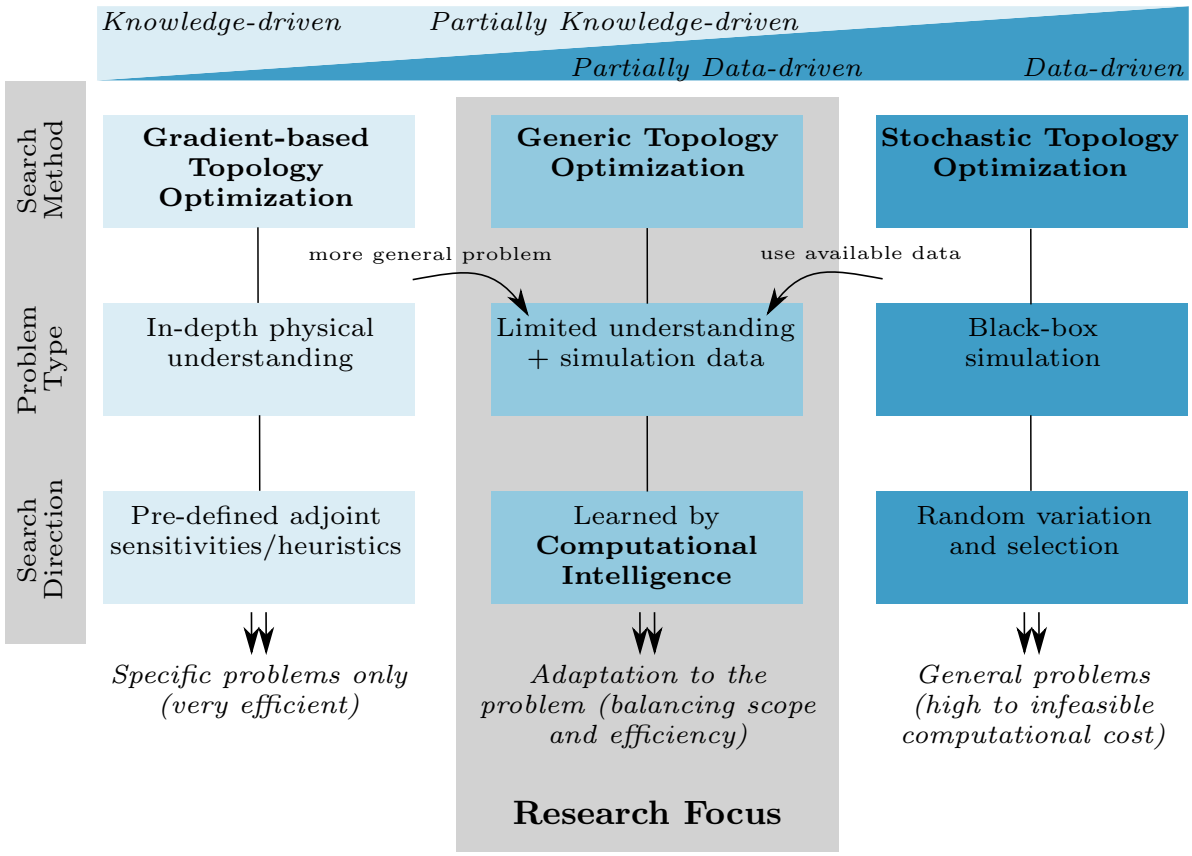


Figure 1.1: Classification of the research focus of this thesis in between knowledge and data-driven approaches. Gradient-based approaches require a theoretical understanding of the considered problem. They are efficient, yet problem-specific. In contrast, stochastic methods, such as evolutionary algorithms, are general, but require high computational efforts. The proposed, generic concept aims at a balance by using simulation data in combination with a computational intelligence method that uniquely adapts the algorithm to the problem.

computational learning procedures. The first approach is based on direct optimization of model parameters with an evolution strategy. Here, the parameters of an artificial, neural network or a piecewise-constant function are optimized. The second approach is to determine the model parameters by a supervised learning method based on finite difference sampling.

Chapter 5 presents a comprehensive assessment of generic topology optimization. The empirical evaluation utilizes a minimum compliance problem that is used as reference. An important aspect of the conducted experiments is the consideration of two different model inputs. Both input types are based on simulation data, but differ in the availability of the local

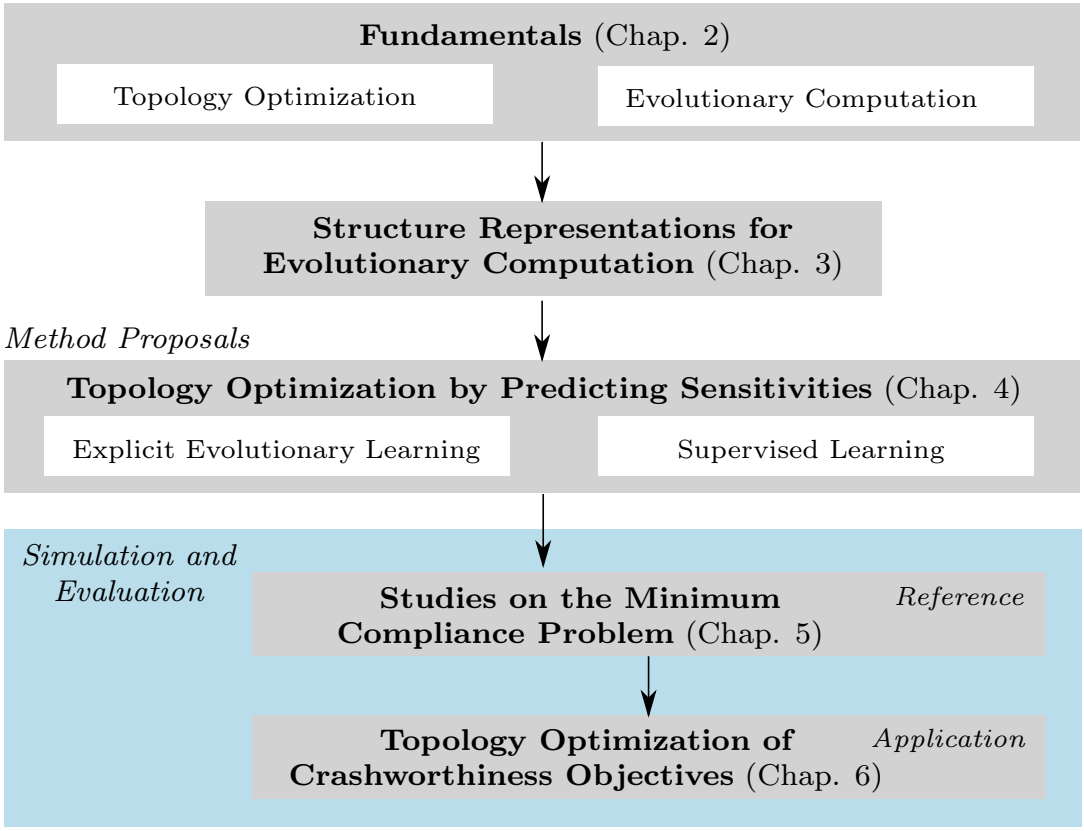


Figure 1.2: Graphical overview of the main components of the thesis.

strain energy density. For most proposed learning methods, the reference structure can be reproduced, even without inclusion of the local strain energy density. Among other things, the various studies also address the effects of model complexity, mesh-dependency and numerical noise. The findings are discussed and lead to recommendations for possible use cases of the methods.

Chapter 6 presents a step towards application. The considered example application is the domain of passive safety in vehicle crashworthiness. Crashworthiness topology optimization is an emerging and challenging task in the vehicle design process. Its implementation in industry has the potential to improve the efficiency of components that undergo extreme loads in crash scenarios. The non-linear physics of crash events yield multi-modal, disconnected and noisy objective functions, which restrict the applicability of the standard topology optimization algorithms. The proposed generic method is applied to optimize two typical objective functions: maximizing energy absorption and minimizing intrusion. It is

compared to an alternative, heuristic method that is applied in industry. The findings support that the proposed method enables the topology optimization of a new class of problems.

Chapter 7 concludes the thesis by summarizing the main contributions and outlining interesting directions for further research.

2 Fundamentals

This chapter provides background on the addressed topology optimization problem and an overview of state-of-the-art optimization methods. Section 2.1 introduces topology optimization of continuum structures, by presenting a general problem formulation, background literature on the field, and an overview of current methods. Section 2.2 elaborates density-based topology optimization methods which are applied throughout this work. The minimum compliance problem is presented as a common test case with examples of optimizations. In contrast to the specialized, gradient-based topology optimization algorithms, Sec. 2.3 introduces the field of evolutionary computation, which provides more general stochastic optimization methods. The chapter is finalized with a summary in Sec. 2.4.

2.1 Background: Topology Optimization of Continuum Structures

In the field of structural optimization, sizing optimization, shape optimization and topology optimization can be differentiated [30]. The three types of structural optimization are illustrated in Fig. 2.1. Sizing optimization refers to optimization of design parameters on a fixed domain that represent, for instance, thicknesses of metal sheets or members in a truss structure. In shape optimization, the design variables describe the form and location of the boundary of a structure. Sizing and shape optimization are commonly applied relatively late in the design process, when a large part of the design is already fixed and optimization is applied as a final tuning step.

Topology optimization can be differentiated into continuum and discrete approaches. Discrete topology optimization tackles the problem of determining which of a discrete set of members ought to be present in a design, e.g., trusses in a truss structure. In contrast, continuum topology optimization calculates the distribution of material within a given design space, resulting in a geometrical layout defined by the shape of void and material regions. In practice, continuum topology optimization

is addressed in discretized form by considering the absence or presence of material within grid elements inside a given design space. This thesis, will exclusively consider continuum topology optimization, although the presented approaches that are discussed hereafter are transferable to other problems, such as truss topology optimization. A fundamental introduction to continuum topology optimization can be found in recent reviews by Deaton and Grandhi [52], or Sigmund and Maute [171], or in the monograph by Bendsøe and Sigmund [30].

2.1.1 General Problem Formulation

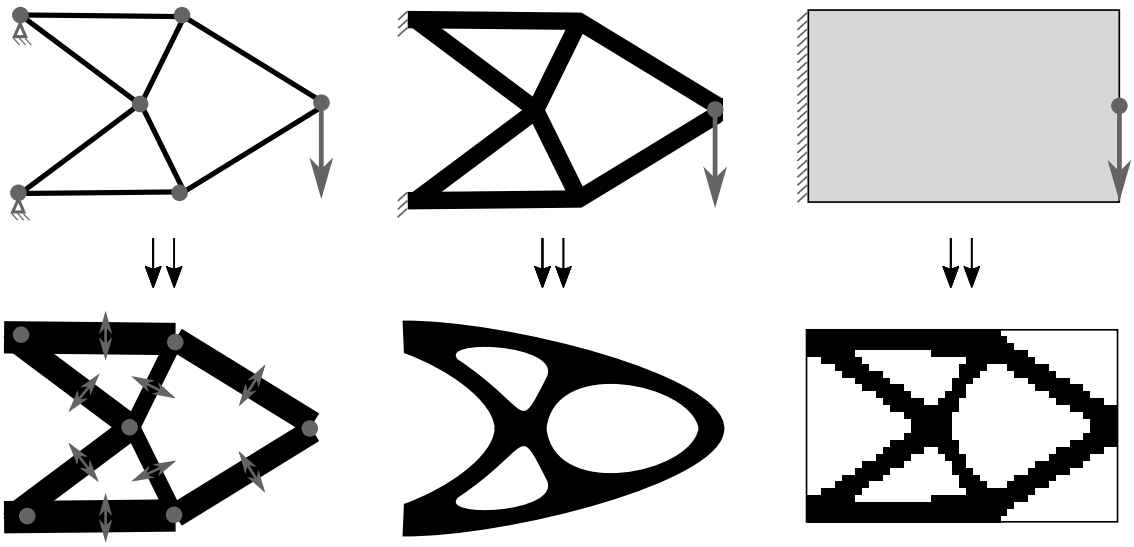
The general topology optimization problem can be stated as [171][1]:

$$\begin{aligned} &\min_{\rho} \; F(\mathbf{u}(\rho), \rho) \\ &\text{s.t.} : G_0(\rho) = \int_{\Omega} \rho(\mathbf{x}) dV - V_0 \leq 0 \\ &\qquad G_l(\mathbf{u}(\rho), \rho) \leq 0, \; l = 1, \ldots, L \\ &\qquad \rho(\mathbf{x}) = 0 \text{ or } 1, \; \forall \mathbf{x} \in \Omega \;\; , \end{aligned} \tag{2.1}$$

where the density variable ρ describes the material distribution at point $\mathbf{x}$ in the design space (or design domain) Ω, $F(\rho, \mathbf{u}(\rho))$ is the objective function, $\mathbf{u}(\rho)$ is the state vector, and G_0 and G_l are optimization constraints with the target volume V_0.

In topology optimization, the term "structure" refers to a design space. Since the boundaries of the optimized structure are unknown a-priori, only the region available for the design of the structure is defined. For each point $\mathbf{x}$ in Ω, the density can either be zero, representing void, or one, representing material, respectively. The density has to be found such that the objective function $F(\rho, \mathbf{u}(\rho))$ is minimized, which quantifies a performance measure of the structural response. Figure 2.2 illustrates a design space Ω with boundary tractions and supports. The state vector $\mathbf{u}(\rho)$ is defined implicitly in the problem-dependent state equation (additionally to (2.1)) and is obtained by a structural analysis. The minimization is subject to a volume constraint $G_0(\rho)$. This constraint restricts the total

[1] Note that, in contrast to Sigmund and Maute [171], we do not assume that the objective function can be computed as integral over a local function, as, e.g., in the compliance case, since this is not true for the intrusion minimization addressed in Sec. 6.3.



(a) Size **(b)** Shape **(c)** Topology

Figure 2.1: Illustration of the three fields within structural optimization [30]. From left to right: size **(a)**, shape **(b)** and topology **(c)** optimization.

amount of material within the design space, which is defined by the difference of the volume $V(\rho) = \int_\Omega \rho(\mathbf{x})dV$ occupied by material and the target volume V_0. Practically, any topology optimization can be subject to constraints $G_l(\boldsymbol{\rho}, \mathbf{u}(\rho))$ that define additional limits on structural properties, for example displacements or stresses.

2.1.2 Early Topology Optimization

One of the first works related to topology optimization has been published by Michell in the beginning of the 20th century, who studied the economical limits of material in frame structures and provided the foundation for optimal reference solutions to some topology optimization problems [118]. Today's (numerical) topology optimization emerged from the field of structural shape optimization in the 1980s. In this field, the surface of the structural design is parameterized, yet the topology e.g., the number and locations of holes, is fixed, and the introduction of new holes is not easily possible. Overall, shape optimization methods were not suited well for introducing topological changes in the structural optimization process. This problem marked starting point for modern topology optimization: the homogenization method, presented in a seminal paper by Bendsøe and Kikuchi [29]. This method addresses topology optimization as a ma-

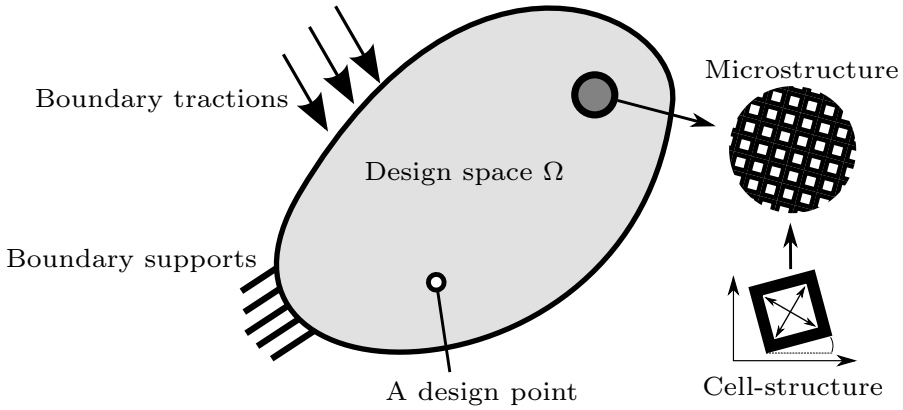


Figure 2.2: General topology optimization problem addressed by the homogenization approach. The material distribution is defined by a microstructure formed by rotatable unit cells with variably sized square holes. As an example for boundary conditions the image shows tractions and supports.

terial distribution problem and considers each design point as consisting of a microstructure defined by geometric parameters of a unit cell. The name "homogenization method" stems from the process of homogenizing the material properties of the microstructures to obtain the corresponding material properties of the macrostructure.

The initial problem addressed by the homogenization method was compliance minimization of linearly elastic structures. An illustration of a topology optimization problem addressed by the homogenization approach is depicted in Fig. 2.2. In the figure, the unit cell of the microstructure is defined by the rotation angle and the size of a square void region. At the maximum size of the void square, a purely void cell is defined. Its minimum describes a filled material cell. For a design space discretized into finite elements, each element is considered as a microstructure.

Succeeding the homogenization method was the density approach, introduced by Bendsøe in 1989 [28], as well as similar approaches by Zhou and Rozvany [217] and Meljnek [120]. The density-based approach, accounts for the fact that the microstructure used in the homogenization approach can be modelled by a material with interpolated properties. The microstructure gets replaced by a porous material, for which a continuous density variable specifies the amount of material at the design point. Material properties are continuously interpolated between void and solid, as illustrated in Fig. 2.3**(a)**. The material interpolation scheme proposed by Bendsøe is based on a power-law approach that penalizes intermediate values of the density with an exponent p [28]. This is shown in Fig.

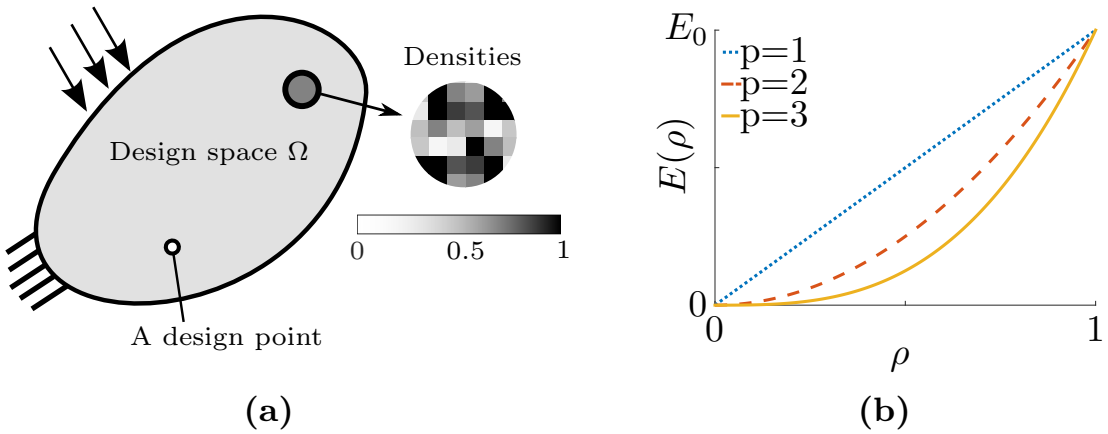


Figure 2.3: General design space of a topology optimization problem addressed by the density approach, where the material distribution is defined by a continuous density variable **(a)**. The SIMP power-law approach to penalize the material properties for intermediate densities **(b)**.

2.3**(b)**. Concretely, the Young's modulus E_0 of the material is reduced for intermediate density values, rendering their use inefficient. This popular approach is know as Solid Isotropic Material with Penalization (SIMP). SIMP has been validated as microstructurally-based model for composite materials for certain values of the penalization [27].

In the 1990s, efficient optimization methods, such as optimality criteria methods [26], the method of moving asymptotes [184, 185], and regularization methods to deal with mesh-dependency [166, 167] in order to achieve numerical stability and existence of solutions in the discretized problem, were established for topology optimization. For linear problems, where gradients are known the mathematical programming methods are able to address also multi-disciplinary problems, combining multiple constraints on compliances, displacements and frequencies [214]

With the advance of numerical methods and increasing computational abilities, the number of topology optimization applications besides the standard topology optimization problem of minimizing compliance[2] rose. Other early applications were, for instance, heat conduction maximization, stress constraints, eigenvalue maximization, buckling, material microstructure design, and compliant mechanisms. An overview of early topology optimization applications can be found in Soto [174]. A later review by Fredericson focuses on application to vehicle body structures [67].

[2]The compliance is the reciprocal value of the stiffness.

2.1.3 Topology Optimization Approaches

Meanwhile, Bendsøe and Sigmund [30] (2004) chronicle comprehensively the development of topology optimization. In the last decade, the field has seen a tremendous development as well as a significant increase in publications. New approaches of optimizing and representing the topology optimization problem have emerged [52, 171]. Approaches can be characterized by the design parameterization, the optimization respectively the update procedure and the regularization method. While a detailed history of topology optimization is not the focus of the present study, considering recent trends in the development of these methods seems useful for gauging the problem from various perspectives. Because topology optimization approaches break up into several subcategories, a brief overview is necessary to illustrate the scientific periphery of the method we propose here. The most common category of topology optimization methods is that of the density-based methods [30], which are described in detail in Sec. 2.2.

The second major field of topology optimization methods, Level Set Methods (LSM), originates from representations used in shape optimization. In LSM [4, 197], the boundary between material and void region is defined by the contours of a level set function. The level set function is typically defined by a combination of (often radial) basis functions. This implicit parameterization of the boundary provides a crisp and smooth interface between void and material region. It facilitates the introduction or disappearance of new holes relative to spline-based shape representations.

A recent review [193] classifies LSM according to their parameterization of the level set function, the geometry mapping, the structural model, the type of update information, and the procedure used. Depending on the LSM variant, shape variations, shape sensitivities, or parameter sensitivities are used in conjunction with mathematical programming methods or heuristic update schemes. A problem that arose for the original LSM approach is that new holes could not be introduced easily in the topology of the structure. Based on the ideas first presented in the so-called bubble method [60], topological derivative methods provide approaches to determine the best point in the structure for the introduction of an infinitesimal small hole and thus, topological changes [3, 36].

Evolutionary Structural Optimization (ESO) approaches directly tackle the discrete topology optimization problem [91]. ESO is similar to

density-based methods in that it parameterizes each element of the discretized design space; however, no relaxation is applied and the variables are treated as binary. The optimization is started from a design space completely filled with material, and the optimization is performed by iteratively removing material elements from the structure, using heuristic update schemes based on a "Sensitivity Number". In an early study, Xie and Steven based the sensitivity number on intuition or heuristic criteria, such as the elemental stress [213]. Meanwhile, state-of-the-art approaches, such as Bi-directional Evolutionary Structural Optimization (BESO), also allow for a re-admission of the material in the elements, while the sensitivity number is based on gradient information [91]. It has to be stressed that the BESO methods are not inspired by Darwinian evolution and have no direct relation to the approaches of evolutionary computation, which we will discuss in Sec. 2.3 and Chap. 3. More precisely, the BESO algorithms do not employ competing and selection properties and thus are not evolutionary algorithms. Therefore, it has been proposed to re-name BESO in Sequential Element Rejections and Admissions (SERA) [145].

Finally, global search algorithms and appropriate representations for topology optimization have also been proposed. These approaches are discussed separately: Section 2.3 introduces the field of evolutionary computation in general and Chapter 3 presents a comprehensive overview and categorization of known representations.

Density-based, LSM, and BESO/SERA are specialized gradient-based methods for topology optimization and represent the most common approaches in the literature. Due to their similarity in terms of gradient-usage all methods provide formulations that could be used as foundation for the method proposed in this thesis. Density-methods provide an excellent balance between efficient and general optimization capabilities, mathematical rigorousness, and intuitive accessibility, and, therefore, will be used as framework throughout the thesis. The next section continues with a detailed introduction to the density-based SIMP approach.

2.2 Density-based Topology Optimization

2.2.1 Problem Formulation

The density-based method is the most common approach for topology optimization that evolved from the homogenization approach. Today, density-based topology optimization tackles a vast number of different applications [52]. The method also achieved wide circulation due to the proliferation of commercial topology optimization software in several key industries. Efficient optimization and regularization approaches for a wide scope of problems are the result of over two decades of research in the field. Other advantages of density-based methods are the - relative to some level set approaches - traceable (yet rigorous) mathematical formulations and the availability of educational resources such as the famous 99- or 88-lines topology optimization codes [5, 168]. Due to these inherent qualities that make the method useful for solving topology optimization for a wide range of problems this thesis follows this approach.

Generally speaking, density-based methods provide a maximum of design freedom within the finite element discretization of the design space, since each element is assigned a variable. The division of the design domain in finite elements yields a discretized formulation of (2.1) [171]:

$$\begin{aligned} &\min_{\boldsymbol{\rho}} \; F(\boldsymbol{\rho}, \mathbf{u}(\boldsymbol{\rho})) \\ &\text{s.t.} : G_0(\boldsymbol{\rho}) = V(\boldsymbol{\rho}) - V_0 \leq 0, \\ &\qquad G_l(\boldsymbol{\rho}, \mathbf{u}(\boldsymbol{\rho})) \leq 0, \; l = 1, \ldots, L, \\ &\qquad \rho_i = 0 \text{ or } 1, \; i = 1, \ldots, N \;\; , \end{aligned} \tag{2.2}$$

where $\boldsymbol{\rho}$ is the vector of design variables and

$$V(\boldsymbol{\rho}) = \sum_{i=1}^{N} v_i \rho_i$$

is the volume of the structure, where v_i is the elemental volume.

Density-based methods relax the discrete problem of which elements should contain void and which elements should contain material into a continuous problem. This enables optimization with gradient-based mathematical programming methods. Concretely, problem (2.2) is relaxed by assigning a continuous "density" variable (or several in case of multi-material problems) $\rho_i \in [\rho_{\min}, 1]$ to each element of the design space. A

minimum value of the density $\rho_{\min}$ is usually introduced to avoid numerical problems that otherwise might occur in finite element solvers, such as close-to-singular matrices. Intermediate densities, however, are undesired in the final solution. Normally, a discrete (zero-one) design is required for manufacturing or further processing. In order to achieve this, the intermediate densities are penalized using a material interpolation scheme. The most frequently used material interpolation scheme applies a power-law approach [27, 28]:

$$E_i(\rho_i) = \rho_i^p E_0 \ , \tag{2.3}$$

with the penalization exponent p, the base material property E_0, and the interpolated material property E_i, see Fig. 2.3b. This is the so-called "SIMP" approach, as introduced in Sec. 2.1.2. Frequently, E_0 refers to the Young's modulus (or elastic modulus) of the material, such as in the important case of the minimum compliance problem that we will consider as a reference case for experiments. However, also other properties can be interpolated.

Using SIMP, problem (2.2) can be formulated as:

$$\begin{aligned} &\min_{\boldsymbol{\rho}} \ F(\boldsymbol{\rho}, \mathbf{u}(\boldsymbol{\rho})) \\ &\text{s.t.} : G_0(\boldsymbol{\rho}) = V(\boldsymbol{\rho}) - V_0 \leq 0, \\ &\qquad G_l(\boldsymbol{\rho}, \mathbf{u}(\boldsymbol{\rho})) \leq 0, \ l = 1, \ldots, L, \\ &\qquad 0 \leq \rho_{\min} \leq \rho_i \leq 1, \ i = 1, \ldots, N \ , \end{aligned} \tag{2.4}$$

where 2.3 effects structural properties such as the stiffness matrix. An example for this is presented for the minimum compliance problem in Sec. 2.2.2.

Since the design variables are continuous in this formulation, gradient information can be used to address the topology optimization problem. Mathematically rigorous terms for the partial derivative of the objective function, with respect to a density variable, are referred to as "*sensitivity*". A typical topology optimization problem has a high number of design variables, but a relatively low number of constraints. Therefore, the formulation of sensitivities is usually addressed by an adjoint analysis [1, 6, 30]. Compared to direct sensitivity analysis, which requires one analysis per variable, the adjoint method reduces the number of analyses to one per constraint and objective. An example of adjoint analysis is presented in App. A.1.

With sensitivity information on objective function and constraints, problem (2.4) can be solved by mathematical programming methods.

State-of-the-art is the Method of Moving Asymptotes, devised by Svanberg [184], or its more recent version the Global Convergent Method of Moving Asymptotes [185]. Many problems can also be solved efficiently by using Optimality Criteria (OC) update methods. OC methods are heuristic update schemes, which increase or decrease the density depending on an optimality criterion, which can be derived from the Karush-Kuhn-Tucker conditions [99, 105]. Although the implementation is problem-dependent and conceptually not suited to handle additional constraints, the OC method is a computationally effective method. It is described in detail in Sec. 4.2.

2.2.2 The Minimum Compliance Problem

The first problem addressed by the topology optimization community was the minimum compliance problem in linear elastics. It refers to minimizing the energy absorbed by the structure when a load is applied, which is equivalent to maximizing the stiffness of a structure. Due to its large proliferation in industrial application, its simplicity, and the availability of reference examples, it is still the most prominent problem in the literature for the evaluation of new topology optimization methods. Chapter 5 includes tests on a minimum compliance reference for evaluating the method proposed in this thesis.

In its finite element form the compliance problem can be stated as:

$$\min_{\boldsymbol{\rho}} c(\mathbf{u}(\boldsymbol{\rho})) = \mathbf{u}^T \mathbf{l} \tag{2.5}$$

$$\text{s.t.} : \mathbf{K}(\boldsymbol{\rho})\mathbf{u} = \mathbf{l}, \tag{2.6}$$

$$\frac{V(\boldsymbol{\rho})}{\sum_{i=1}^{N} v_i} = f, \tag{2.7}$$

$$0 < \rho_{\min} \leq \rho_i \leq 1, \ i = 1, \ldots, N \ , \tag{2.8}$$

where $\mathbf{l}$ is the global load vector, $\mathbf{K}$ is the global stiffness matrix, and (2.6) is the linear static governing equation. The classic compliance problem considers no constraints except for the volume constraint (2.7) and the limits of the density (2.8). The desired ratio of material to void within the design space is the target volume fraction f. The finite element analysis refers to a matrix inversion to obtain the state vector $\mathbf{u}$. The matrix $\mathbf{K}(\boldsymbol{\rho})$ is constructed by superposing elemental stiffness matrices $\mathbf{K}_i(\rho_i)$, in which the Young's modulus is replaced according to the SIMP interpolation (2.3). Therefore, the structure properties and, thus, the stiffness matrix is a

function of the density:

$$\mathbf{K}(\boldsymbol{\rho}) = \sum_{i=1}^{N} \mathbf{K}_i(\rho_i) \ .$$

Since volume is the only constraint in this case, the problem formulated in (2.5)-(2.8) can be solved efficiently an OC-update method.

The structural performance profits from the introduction of smaller and smaller holes, so that the optimum structure would have an infinite number of infinitesimal small holes. In the case of the discretized topology optimization problem, the finer the mesh resolution the smaller the sizes of the optimal structural members become. This causes a problem known as mesh-dependency. The underlying reason is that the topology optimization problem (2.1) is ill-posed, as it "*lacks existence of solutions in its general continuum setting*" ([30], p. 30). Additionally, numerical instabilities such as chequerboard patterns can occur in the solutions [56]. Regularization methods impose a minimum length scale of the structural members and alleviate numerical instabilities [167, 169]. A heuristic, but frequently applied method is filtering of sensitivities [166]. In this method, which is derived from image processing, the sensitivity of an element is averaged based on the sensitivities of neighbouring elements.

The computational flow of a density-based topology optimization algorithm can be illustrated as in Fig. 2.4. The algorithm starts with an initial solution, usually a homogeneous distribution of material over all elements in the design space i.e. $\rho_i = f, \forall\, i = 1, \ldots, N$. This initial structure is analyzed by solving the state equation via a finite element analysis, providing the structural state and thereby the objective function value. In the next step, sensitivity information is computed. Sensitivity information enables us to modify the design by using the chosen update and regularization methods. This loop is then repeated until the optimization has converged.

By way of illustration, the following two classical plane stress design problems are based on a two-dimensional, rectangular design space. Example one is the design of a cantilever beam. Figure 2.5**(a)** depicts the design space and boundary conditions. It is supported in both coordinate dimensions along the left design space boundary. A load is applied in the centre of the right design space boundary. The target volume fraction is set to $f = 0.4$. The design space is meshed with $N = 45 \times 28 = 1{,}260$ square shell elements. Example two is a MBB[3] beam that is supported

[3] Messerschmitt-Bökow-Blohm

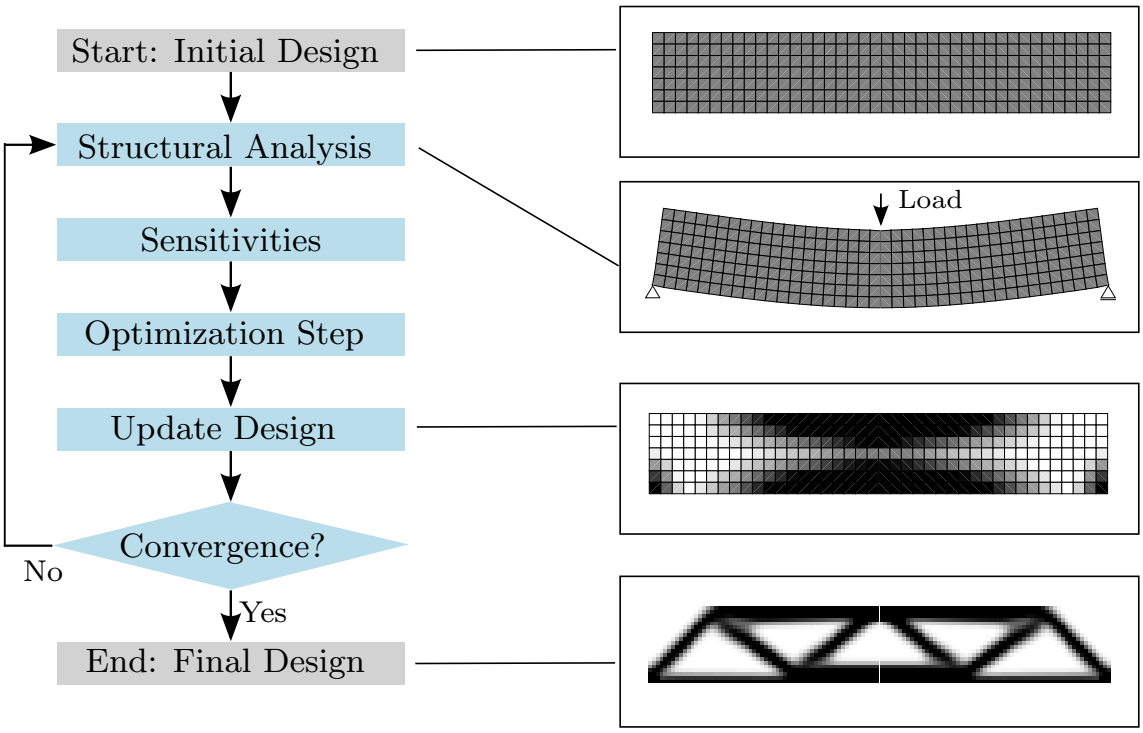


Figure 2.4: Computational flow for density-based topology optimization.

vertically along the left design space boundary to enforce symmetry. It is also supported in the right bottom corner in both coordinate directions, as depicted in Fig. 2.5**(b)**. The load is applied in the upper left corner of the design space, i.e. the centre of the beam. The volume fraction is set to $f = 0.5$. The design space is meshed with a resolution of $N = 150 \times 50 = 7{,}500$ square shell elements. In both cases, the load vector has a unity component at the load node, and is otherwise zero.

Solving the problem formulated in (2.5)-(2.8) with the OC-update and sensitivity filtering results in the optimized structures[4] depicted in Fig. 2.5**(c)** and **(d)**. A comprehensive description of a Matlab implementation of this topology optimization algorithm, including educational code, is available in the excellent paper by Sigmund [168] (see also the more recent, extended and optimized version [5]).

Since the method is based on gradient information, it provides a way to find locally optimal solutions. For a large number of applications this works very effectively. For instance, in a typical density-based topology optimization, such as the presented one, often only a few dozens of iterations are required, even for problems with tens or hundreds of thousands

[4]Sensitivities and parameter settings correspond to those used for the reference test case in Sec. 5.1.

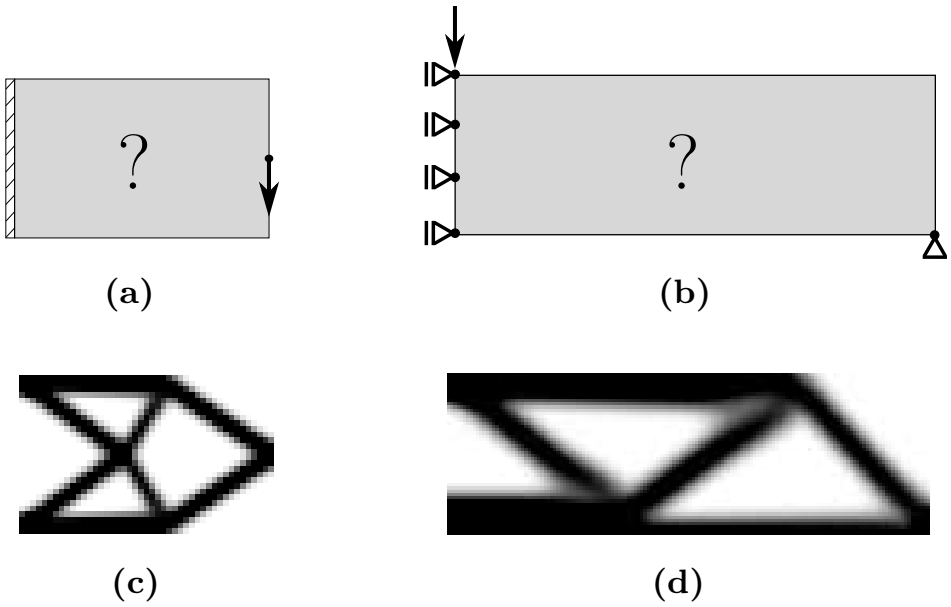


Figure 2.5: Examples for topology optimization: Design space for a cantilever beam **(a)** and solution for 45×28 mesh **(c)**; Design space for a MBB beam **(b)** and solution for 150×50 mesh **(d)**.

variables. Even though the search methods are local, satisfying solutions for many problems can be found. In practice, engineers use topology optimization early on in the design process to produce initial, conceptual designs which provide a starting point for the next design step. At this stage of the design process, the mathematical global optimum is of secondary interest. Usually, the solution of the topology optimization undergoes subsequent post-processing steps, such as manual modification and/or shape or sizing optimizations. For industrial users, it is therefore often sufficient to provide a rough topological layout as a starting point for further work.

However, a necessary requirement for the application of the algorithms presented here is gradient information. When derivatives are not available, the established algorithms cannot easily be applied. Additionally, other characteristics of objective functions, such as non-linearity, ruggedness, multi-modality or noise, can reduce the success of gradient-based algorithms. Stochastic search algorithms from the field of Evolutionary Computation are independent of gradients and therefore appear suitable to provide a work-around for this deficiency. This field and its search algorithms are introduced in the next section.

2.3 Evolutionary Computation

Compared to non-gradient approaches, gradient-based topology optimization methods are significantly more efficient with respect to function evaluations [170]. However, the most obvious restriction is the necessity of a gradient. In practical design optimization problems, explicit mathematical modelling can be difficult due to problem complexity or black-box simulations. Also, the determination of numerical gradients can be rendered infeasible by numerical noise or a high number of variables. Furthermore, for problems with severe non-linearities that result in highly multi-modal, rugged, or discontinuous search spaces, a gradient-based search is likely to yield non-satisfying results[5].

This chapter describes the field of Evolutionary Computation (EC) [19, 59] as an alternative. This field encompasses a large variety of derivative-free search algorithms with stochastic component that can be applied to general optimization problems. It is thus also suitable for addressing the shortcomings of conventional topology optimization methods. The biological foundation of EC is the theory of evolution based on natural selection [47] that revolutionized the understanding of biological life. It reveals biological life as an ongoing adaptation process of species for survival. Accordingly, the history of organic life can be understood in terms of physical processes of adaptation, which are reproduction, variation and selection. In the nineteen sixties, this inspired a number of computational approaches to mimic the evolutionary processes. Whenever these mechanisms are at work, an evolutionary process is inevitable. Thus, by conceptually implementing these mechanisms in a computer program, evolution can be simulated and utilized as a model for addressing technical problems or for improving human understanding of the gradual, adaptation processes that shape the development of life (e.g., [49]).

The field of Evolutionary Computation, as the intersection of evolutionary biology and computer science, started developing in the 1960s. Initially, three main types of Evolutionary Algorithms (EA) could be distinguished: Evolution Strategies, Genetic Algorithms, and Evolutionary Programming. Here, the abbreviations EC and EA roughly stand for the whole research field and for the related algorithms, respectively, although there is certainly some overlap between the two terms.

Evolution Strategies were first developed for real-valued parameter opti-

[5] An example is the area of crashworthiness topology optimization, for which strong assumptions for model simplifications have to be made, in order to perform optimization, see Sec. 6.1 for details.

mization of a wing shape [139]. This very practical inspiration contributes to explain why evolutionary strategies today are applied to many design optimization tasks. Genetic Algorithms were inspired by knowledge gathered on the genetic code [84]. Solutions are encoded in genes that are represented by an array of discrete values. New solutions are created by recombination, a method that is inspired by the genetic crossover occurring in biological life during sexual reproduction. Evolutionary Programming [64] is based on the idea of evolving functions or programs that yield a certain output when executed.

Nowadays, EC forms a large lively field of research and boundaries between the three groups of algorithms are blurred. Operators and algorithms have been unified in a common framework, e.g. [20]. In the context of optimization, evolutionary algorithms are a class of derivative-free search algorithms [143], well suited for noisy, multi-modal, disconnected, non-separable, dynamic or black-box problems, that often occur in real-world engineering design optimization. Due to their robustness and generality they are a good choice for problems where specialized methods do fail or do not exist in the first place. These advantageous properties, however, come at high computational cost, since no optimization method can be general and effective at the same time(a fact known as no-free-lunch theorem [208]).

The concept of a typical evolutionary algorithm is illustrated in Fig. 2.6. In the field of EC, usually biologically-inspired terminology is applied, i.e. a set of solutions is referred to as a population of individuals. A starting point for the optimization is a population consisting of one or several individuals. Initially, these individuals are evaluated to determine their objective value or the "fitness". A new "offspring" population is created by variation operators that mimic natural, generational processes such as random mutation and sexual recombination. Based on the fitness of the individuals, a selection operator chooses a subset of individuals as the parent population for the next generation. The evolutionary optimization loop is stopped, when a predefined criterion is reached, for instance, the change of the fitness drops below a threshold or a maximum of evaluations is reached.

EAs are relatively robust and general. They have the potential to identify global optima and do not require gradient information. They can also address problems with strong non-linearities or black-box simulations, while having the disadvantage of considerably higher computational demands. On the plus side, EAs only require implicit assumptions about the objective function, e.g., the choice of representation and variational op-

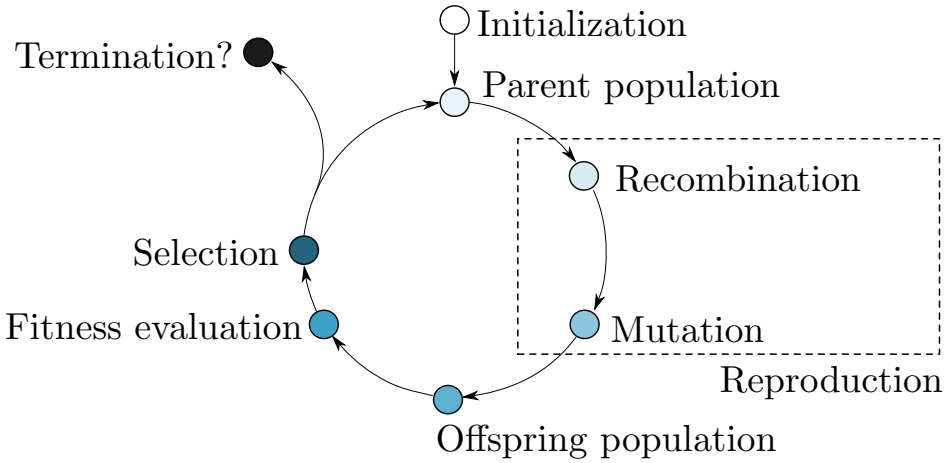


Figure 2.6: The working principle of an evolutionary algorithm.

erators. This universality is an important advantage over gradient-based methods. Throughout this thesis, EAs are not proposed as direct competitors to gradient-based approaches. Rather they are considered as an alternative for especially difficult problems, in which gradient-based methods do not provide satisfactory results. We focus now on the field of Evolution Strategies, which are applied later on in this work.

2.3.1 Evolution Strategies

One specialization of evolutionary algorithms are Evolution Strategies (ES). This area of studies is based on the seminal work done by Rechenberg and Schwefel who optimized the shape of a wing in a wind-tunnel experiment [139, 140, 158]. Initial, gradient-based variation of the shape parameters yielded trivial flat solutions. Only when a stochastic component was introduced, did the optimization succeed. Throughout the study, the parameters describing the shape were modified by random numbers obtained from a normal distribution. In case the performance of the structure improved, the changes were accepted. The algorithm was later classified as the first approach towards an ES.

Background and introductions on ES can be found in the literature [21, 31, 146, 159]. Unbiased, deterministic selection, a parameterization of the mutation operator, and the fact that individuals consist of design variables and strategy parameters are characteristic for ES. The recombination operator creates new offspring individuals from the existing solutions. It

either applies intermediate or discrete (dominant) recombination of properties, yet is considered to play a minor role in typical ES. Variation of the individuals is mainly performed by the mutation operator, which provides random numbers from normal distributions to modify design and strategy parameters and predominantly determines the search.

An ES is defined by the number of parent individuals μ, the number of offspring individuals λ, the number of parent individuals for the creation of a new offspring individual ϱ and the lifetime κ, i.e. the number of iterations an individuals can stay in the population. In the most cases, the lifetime is either infinite or one, resulting in a more simplified notation: If the novel parents are chosen exclusively from the offspring population, i.e. $(\mu/\varrho, \lambda)$-ES, it is a "comma" strategy; if they are chosen from parent and offspring populations, i.e. $(\mu/\varrho + \lambda)$-ES it is a "plus"or "elitist" strategy. In ES, the parents for recombination are usually selected in a deterministic way.

Furthermore, an ES is defined by the variation operator and its finite set of parameters. Since in ES these parameters can be subject to change during the optimization, they are also denoted as endogenous strategy parameters.

Favourable properties of the mutation operator are the ability to reach any point in the search space, unbiasedness, and the possibility of parameterization. These desirable properties foster mutation based on a multivariate, normal distribution, i.e., the vector of design variables $\boldsymbol{\theta} \in \mathbb{R}^{\Theta}$, encoded by an individual is mutated by[6]:

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \mathbf{z},\ \mathbf{z} \sim \mathcal{N}(0, \mathbf{C}) \quad ,$$

where k is the generation (i.e., iteration) of the optimization, $\mathbf{z}$ is a random vector drawn from the multivariate normal distribution $\mathcal{N}(0, \mathbf{C})$ with covariance matrix $\mathbf{C} \in \mathbb{R}^{\Theta \times \Theta}$. Since in a typical ES, the design variables are real-valued, ES are especially applicable for continuous optimization problems as occurring frequently in real-world engineering design optimization.

In ES, this allows for three main possibilities for the mutation:

- A single global step size is used: $\mathcal{N}(0, \mathbf{C}) \sim \sigma^2 \mathcal{N}(0, \mathbf{I}_{\boldsymbol{\Theta}})$, i.e. there is one strategy parameter. Here, $\mathbf{I}_{\boldsymbol{\Theta}}$ is the identity matrix.

[6] The evolutionary optimization and the topology optimization are addressed further in Chap. 4, where both are combined in a nested scheme. In order to consistently distinguish between both optimization loops, a different notation than the one described in the extant EC literature is introduced. In the context of topology optimization, the design variables are denoted as $\rho_i,\ i = 1 \ \ldots \ N$, while in context of the evolutionary optimization, the Θ design variables are denoted as $\theta_i,\ i = 1 \ \ldots \ \Theta$.

- For each dimension of the search space a separate step size is used. In this case, $\mathbf{C}$ is a diagonal matrix consisting of the standard deviations σ_1^2, σ_2^2, $\ldots, \sigma_\Theta^2$, i.e. there are Θ strategy parameters.
- The full covariance matrix is used, hence there can be correlated mutations. This results in total in $\Theta(\Theta+1)/2$ strategy parameters.

The strategy parameter σ is called the "step size" or "mutation strength".

The ideal strategy parameters not only depend on the objective function, but for most problems also change during the optimization. In early research on ES, investigations by Rechenberg with the $(1+1)$-ES on the sphere and corridor test functions, provided the insight that the ideal probability of successful mutations should be 0.2, which lead to the 1/5th-rule [140]. According to the 1/5th-rule, the step size is decreased when less than one-fifth of the mutations is successful, and increased if more than one fifth is successful and remains unchanged otherwise.

However, the 1/5th-rule is only optimal in the special case of a single global step size in a $(1+1)$-ES,. This leads to the idea of self-adapting strategy parameters that can be considered a standard concept in ES [31]. In a self-adaptation ES, an individual consists not only of the design variables, but also of strategy parameters. In a standard self-adaptation ES, the recombination operator is applied first, combining ϱ parents to a new offspring individual. Typically, discrete recombination is used for design parameters and intermediate recombination is recommended for strategy parameters [31].

Hence, step size or, in the more general case, the elements of the covariance matrix are themselves subject to selection. Implicitly, by selecting the best individuals, on average also suitable mutation parameters are selected, so that the strategy parameters of individuals adapt to the local search space. For instance, let us consider the case of diagonal elements of the covariance matrix. Then, a vector of strategy parameters $\boldsymbol{\sigma}$ per individual is obtained, which means that there is one strategy parameter per design variable. The mutation of each component is performed multiplicatively according to:

$$\sigma_i^{(k+1)} = \sigma_i^{(k)} \cdot \exp(z_0) \cdot \exp(z_i),\ z_0 \sim \tau_0 \mathcal{N}(0,1),\ z_i \sim \tau \mathcal{N}(0,1)\ ,$$

where $\tau_0 = \frac{1}{\sqrt{2\Theta}}$ and $\tau = \frac{1}{\sqrt{2\sqrt{\Theta}}}$ are recommended learning parameters.

However, with self-adaptation, it is still possible to select individuals that have performed a large step, although the step size is small or vice

versa. Also, the large number of strategy parameters, especially for correlated mutations when the full covariance matrix is considered, requires a high number of evaluations until adequate adaptation is reached. Reducing the impractically high number of evaluations makes it necessary to maximize the benefit from successful mutations. This issue leads to the concept of the Covariance Matrix Adaptation Evolutionary Strategy.

2.3.2 Covariance-Matrix Adaptation Evolutionary Strategy

With increasing number of strategy parameters, the self-adaptation of the diagonal elements or even rotation angles of the covariance matrix requires an increasing learning effort that deteriorates the efficiency of the algorithms. Therefore, most modern ES try to systematically estimate a distribution from which favourable solutions can be drawn. With this respect, the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [78] is considered as state-of-the-art [21]. The CMA-ES adapts the covariance matrix of the multivariate normal distribution, from which new solutions are drawn, based on successful mutations, hence a model of the underlying fitness landscape is learned. By this means, the effect of randomness is reduced, therefore, the CMA-ES is also referred to as a "derandomized" evolution strategy. In this section, a conceptual description is provided based on the CMA-ES by Hansen [76]. For a detailed, formal description the reader is referred to the original sources.

In the CMA-ES the new offspring population is created based on sampling a multi-variate normal distribution:

$$\boldsymbol{\theta}^{(k+1)} \sim \mathcal{N}\left(\mathbf{m}^{(k)}, \left(\sigma^{(k)}\right)^2 \mathbf{C}^{(k)}\right) \ ,$$

where $\mathbf{m} \in \mathbb{R}^{\Theta}$ is the mean value of the search distribution. It is based on the weighted, ranked intermediate recombination of the μ best individuals.

The covariance matrix is adapted iteratively, based on the following main concepts:

- The matrix is estimated based on the previously selected steps, hence it estimates the distribution of *successful* mutations (in contrast to estimating the distribution of the population or the distribution of mutations).
- For small population sizes an acceptable estimate of the covariance matrix cannot be obtained from one iteration. Hence, for the update

of the covariance matrix, information from the previous iterations is taken into account, subject to exponential smoothing.

- Information from a cumulation of several successive steps of the optimization that forms an evolutionary path, is included. Exponential smoothing of the evolutionary path is applied and it is used as an additional term in the update of the covariance matrix. This enables learning about correlations between steps, for instance about directional information contained in the signs of the mutations that would be lost otherwise.

Additionally, the global step size is controlled by a method denoted as cumulative path length control. Similar to the covariance matrix update it is based on consecutive iterations i.e. the evolutionary path. For adapting the step size, the length of the evolutionary path is evaluated with respect to the expected path length under random selection. In case the path length is longer than expected, i.e. selection biases towards larger steps, the step size is increased, otherwise it is decreased.

Favourable properties of the CMA-ES involve several invariances with respect to the objective function, for instance invariance towards order-preserving and angle-preserving transformations as well as scaling and unbiasedness of the variation. Only initial solution and initial global step size need to be chosen, and an almost parameter-free ES results, that shows good to superior performance on a large number of test functions and has been applied to a large number of applications [21, 76–79, 141].

Compared to other EAs, an advantage of the CMA-ES is the relatively low number of function evaluations that is required in order to obtain decent optimization results. Since the evaluation is usually the most computationally expensive step in a real-world optimization, this renders the CMA-ES an effective method and facilitates the application in industrial contexts, where often only a few hundreds or thousands of evaluations are feasible [21]. Several extensions of the CMA-ES have been proposed; for instance, local restarts [7, 8] as well as the Active-CMA-ES that also actively uses the individuals with lowest fitness in the covariance adaptation step [95].

2.4 Summary

Within this chapter, the necessary background on literature and fundamental theory in the fields of topology optimization and evolutionary

computation is described. At first, the general continuum topology optimization problem is introduced as a material distribution problem, in conjunction with its typical solution methods. Special emphasis is placed on the density-based formulation as a very efficient and general approach. Yet, as other standard methods, it depends on derivative information, which is unavailable for many real-world and especially black-box problems. Evolutionary computation provides a viable alternative approach because it features general-purpose optimization algorithms that have been well established in scholarship and practice. Within this field, evolutionary strategies are particularly relevant for the innovative optimization approach proposed later on. However, one of the most important aspects when performing topology optimization with an evolutionary computation approach is the representation of the structure, which is why the following chapter includes a comprehensive review as well as a categorization of existing representations.

3 Structure Representations for Evolutionary Computation

About a decade ago, Kicinger et al. provided an overview on Evolutionary Computation (EC) and structural design, including approaches for continuum topology optimization [101]. Since then, a large variety of specialized methods has been developed. However, researchers in structural design are often not aware of the available EC methods, while researchers from the EC community are often unfamiliar with the topology optimization problem. Bridging this gap helps to broaden the range of available optimization approaches and as a result the range of problems that can be addressed. This chapter contributes to the discourse by categorizing and by summarizing existing representations for EC approaches. We define three fundamental classes of representation with different advantages and drawbacks.

Section 3.1 discusses the subject of the representation problem and the different classes of representations. In the next step, the different representations are described, starting with the Grid Representation in Sec. 3.2, followed by the Geometric Representation in Sec. 3.3, and finally the Indirect Representation in Sec. 3.4. The different representations as well as assets and challenges of EC for topology optimization are summarized in Sec. 3.5. Parts of this chapter are based on [11].

3.1 Representing the Structure

Initial EC approaches in the area of continuum topology optimization arose briefly after the introduction of the homogenization method. One of the first applications was the optimization of the compliance problem using a genetic algorithm. Thereby, every element of the discretized design space is represented by a variable [42, 147]. Today, the literature comprises numerous evolutionary methods with varying representations. In view of

the high computational demand of EC methods, an especially important aspect of any application of an evolutionary optimization algorithm is the parameterization of the problem. In the following paragraphs, we briefly describe the concept of geno- and phenotype for representing solutions and relate it to topology optimization.

In biology, the genotype refers to parts or the complete genetic code of a cell or organism. The phenotype refers to the actual expression of the genotype as the living individual (e.g. a finch) or a characteristic (e.g. beak) that is exposed to the environment. Although the phenotype is what is exposed, indirectly its encoding genotype is subject to natural selection [48]. Similar to the biological inspiration, in the field of EC, the phenotype is the solution in the context of the original problem that is subject to fitness evaluation. It is described by its genotype, to which variational operators are applied. Hence, the actual optimization is performed on the level of genotypes. Eiben and Smith [59] define "*representation*"(or encoding) as "*mapping from the phenotypes onto a set of genotypes*".

Important representations are binary encodings used in genetic algorithms; for instance, when a continuous number is represented as a binary number. A more natural encoding is that of a binary encoding for decision problems such as the Travelling Salesperson Problem. Another natural encoding are real-valued design variables in evolution strategies; for instance, for the purpose of engineering form optimization problems.

The representation not only determines the dimensionality of the search space, but also influences the probability of the identification of improved solutions and consequently the progress rate of the optimization. Often, a well chosen representation can reduce the complexity of the original problem. Representations themselves can be complex, for instance, by involving developmental steps with a small number of parameters that encode most versatile phenotypic variation (see for instance [49]). Hence, depending on the representation, the optimizer has to deal with different search spaces. A well-chosen representation facilitates beneficial changes obtained from search operators. This is related to the topic of evolvability in the field of EC, which captures the ability of a representation to efficiently improve solutions [12, 50, 88]. Most important, the representation should be able to include the optimal phenotypes, or, at least, close to optimal versions.

EAs are global search algorithms and have a chance to locate the global optimum. With respect to topology optimization, they are able to provide a solution directly applicable to the discrete problem in (2.2). Since the evaluation of the structure requires a finite element analysis, the structure

is discretized into a finite element mesh. Following this argumentation, the phenotype in structural optimization is formed by a grid of finite elements that is subject to evaluation. However, searches based on a direct bit-wise representation of the elements are not efficient for fine grid resolutions, since the number of possible solutions increases exponentially with the number of elements. More elaborate representations reduce the number of design variables, but they naturally also reduce the set of possible solutions. In conclusion, the application of an appropriate representation for topology optimization with EAs is an important first step, which fundamentally influences the computational feasibility.

This chapter proposes three concepts of representations, which are used in the field of EC: Grid, Geometric and Indirect Representation. Figure 3.1 shows an overview of the proposed representation classes. These classes allow for a categorization and discussion of the existing representations, which are used in conjunction with EC, for structures in continuum topology optimization.

3.2 Grid Representation

A representation can be classified as "*Grid Representation*", if the genotype encodes properties of fixed locations in a grid that discretizes the design space. Typically, the number of search variables is identical or proportional to the number of grid cells. A maximum of design flexibility is realized, when a variable is assigned to each cell of a finite element mesh discretization of the design space.

3.2.1 Bit-Array Encoding

The most straightforward form of the Grid Representation is the bit-array encoding. The genotype encodes a binary variable for each grid cell, representing either void or material. This is illustrated in Fig. 3.2. In the most common case, each variable of the grid cell corresponds to material in one finite element of the analysis model.

An early work that optimizes a bit-array representation by using a genetic algorithm, minimizes the weight of a cantilever plate, subject to displacement and stress constraints [147]. This approach requires a connectivity analysis: material elements that do not share an edge with at least one neighbouring material element or that are not directly or indirectly connected to the boundary conditions are switched to void. The

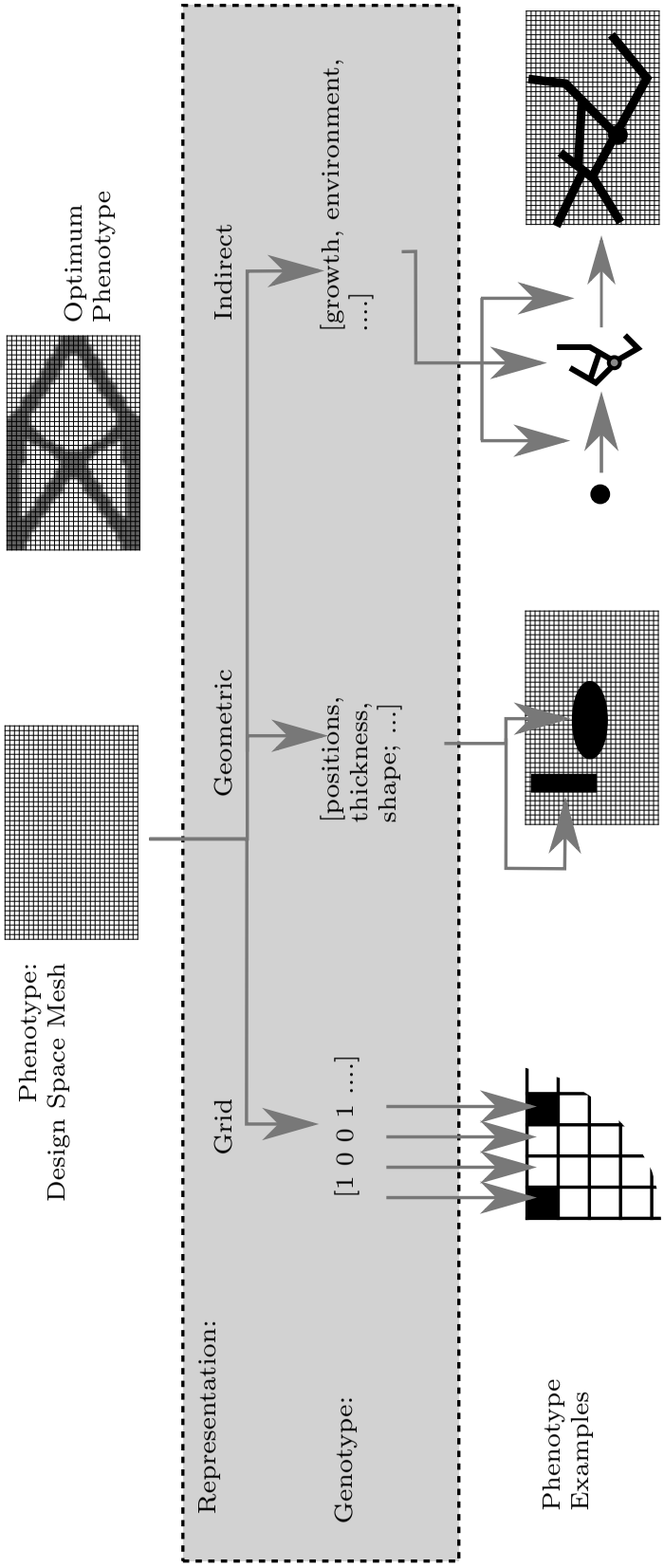


Figure 3.1: Proposed categories of representations for structures in a topology optimization scenario.

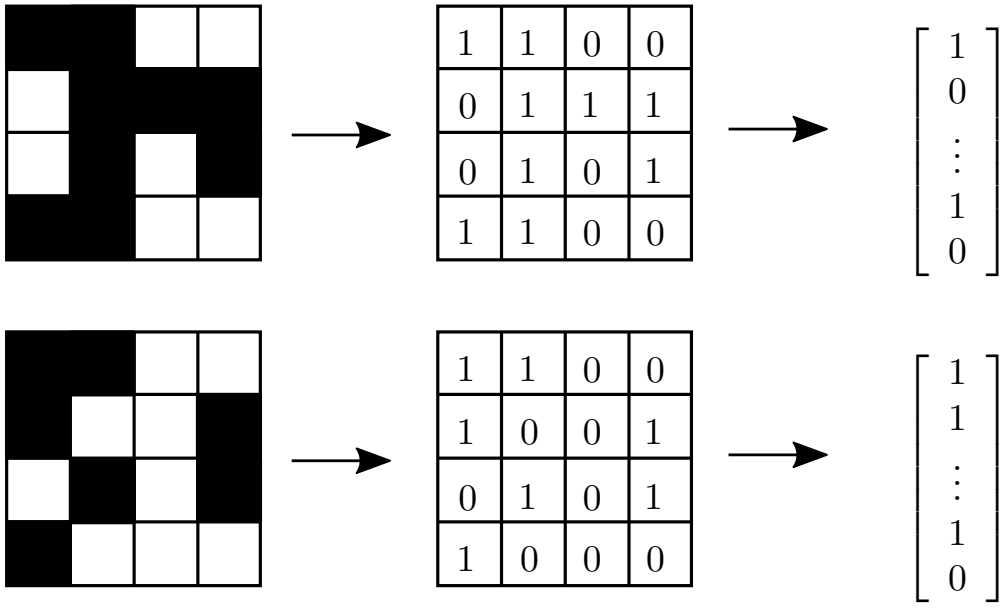


Figure 3.2: Example of a Grid Representation with bit-array of a connected structure (top). Each element is represented as one binary variable, which enables a maximum of design variability. This however includes undesirable solutions, for instance unconnected structures (bottom).

bit-array representation has been extended for the application to slightly more finely discretized solutions by introducing a hierarchical design domain subdivision [42]: After optimizing based on a coarse finite element representation, the design space is subdivided to refine the optimized solution and the optimization is continued with several populations.

Plenty of different domain-specific optimization approaches have been proposed since then, based on genetic algorithms [22, 43, 103, 196, 203, 205], artificial immune system algorithms [37, 111], particle swarm optimization [110, 112] or modified binary differential evolution [211]. Multi-objective genetic algorithms combined with local search operators have been applied [161, 162], as well.

Some approaches combine EC with state or gradient information. In [164] a simulated annealing is combined with information on the elemental stress. In [107] the solutions are recombined based on the stress contours of the design in order to guide the optimization. Furthermore, an ant-colony optimization algorithm is modified to include gradient information by biasing the pheromone concentration based on elemental sensitivity information. The algorithm is applied e.g. to compliance minimization [100] and compliant mechanism design [102]. Using the results of gradient-based methods for population initialization, a multi-objective topology

optimization method is proposed in [113, 114].

Although technically based on a bit-array representation, the so-called Evolutionary Structural Optimization (ESO) approaches [89, 213] do not include any concept of population, random variation or selection and thus do not classify as evolutionary algorithm as discussed in Sec. 2.1.3.

3.2.2 Real-Valued Array Encoding

When operating directly on the mesh, gradient-based algorithms [30] apply a continuous variable to each mesh element, due to the necessity of gradual changes. This is rarely done in the existing EC approaches, most likely since genetic algorithms are able to directly address the discrete problem.

An exception is the work of Madeira et al. in which each element of the mesh is assigned a real-valued cost variable [115]. Each element is defined as node in a graph, connected by edges between neighbouring elements. Still, the representation involves one variable per element in the mesh, hence, it classifies as a Grid Representation. The problem of finding the optimum structure for a volume constraint is transformed to the task of finding the optimum subtree, including the correct number of vertices with the lowest cost. An advantage of the method is that the subtree finding process only generates connected structures and crossover and mutation to the cost variables with a genetic algorithm is straightforward.

3.3 Geometric Representation

A representation can be classified as "*Geometric Representation*", if the genotype encodes properties of a set of movable shape primitives that define the geometry of the structure. Properties of the shape primitives are for instance position, shape and thickness. When decoding, a geometry mapping is applied to obtain the phenotypic finite element mesh. The search space dimensionality as well as the potential structural complexity depend on the number of primitives, but both are independent of the number of finite elements used in the structural analysis.

3.3.1 Voronoi-cells

Motivated by the idea of overcoming the direct coupling of the finite element mesh and the number of variables, Schoenauer described a geometry for a topology optimization by Voronoi cells [155]. The genotype encodes

a set of coordinates for the Voronoi sites. Voronoi sites divide the design space into polyhedral-shaped subsets, termed Voronoi cells, in which each point of the subset is assigned to the nearest Voronoi site. Every Voronoi site is assigned a binary variable, i.e. "material"or "void". This approach is capable to represent the geometry of any structure formed by polyhedral subsets, and to map it to the underlying structural model. The illustrations in Fig. 3.3 depict the Voronoi representation and a structural mapping. The evaluation has been conducted using genetic algorithms and evolutionary strategies [155]. The idea was picked up and applied by several researchers [53, 54, 72, 73].

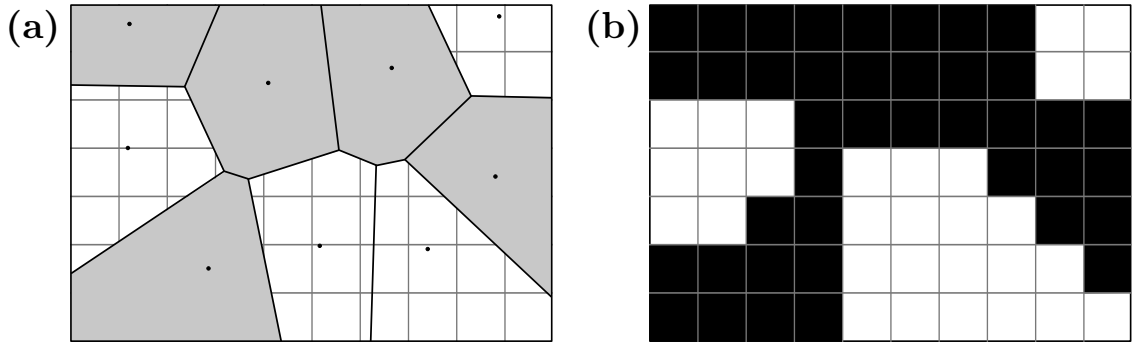


Figure 3.3: Example for **(a)** geometric form defined by a Voronoi representation of the structure and **(b)** the corresponding structural design, as in [72].

3.3.2 Material-Mask Overlay

Schoenauer also proposed a representation based on optimizing the position and shape of geometric forms [155]. The work considers rectangular holes in the design space, which is otherwise filled with material. The approach can be considered as an early version of the material-mask overlay strategy proposed in [94, 129, 150, 151], where a set of circles is optimized. For each circle/mask, the Cartesian coordinates of its centre and radius, and its material state are encoded as variables. A user-defined number of masks is mapped to the structure by assigning material or void to the elements of a hexagonal element mesh based on the mask material state. The material state is a binary variable. Elements whose centres lie within the radius of a mask are turned into material or void elements respectively. If masks overlap, the topmost mask decides whether an element in the intersection is filled with material or void. Thereby, the number of masks is an important parameter, which decides on the search space dimensionality

and implicitly on the structural complexity that can be represented. Optimization is performed with a multi-objective particle swarm optimization. An adaptive variant has been proposed, in which the number of masks is adapted during the optimization, by removing redundant masks [152]. Furthermore, only negative i.e. void masks are applied so that material is by default put in all regions of the design space without masks. Figure 3.4 shows an illustration of the concept. Parallel, mutation-only search is applied. Other geometric masks such as elliptic or rectangular ones may be used as well, although circular masks seem to be more efficient[153]. The efficiency of the method can be further increased by using gradient information [154].

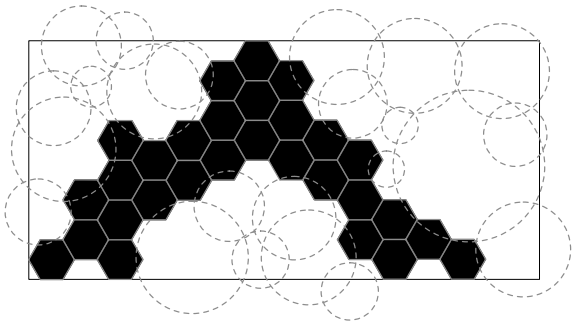


Figure 3.4: Representation with Material mask overlay strategy, each mask (circle) defines a location where material is removed from the design space that is otherwise filled with material elements [150].

3.3.3 Graph Representation

In contrast to having independent components, several researchers have explored the idea of representing the phenotype as a graph. In these cases, the genotype encodes positions of nodes and characteristics of the edges such as form or thickness. Examples for this are graphs in which edges represent beam elements or spline curves that are connected through the nodes.

Bézier Curves

Tai and Chee connected locations, where the structure interacts with boundary conditions such as loads and supports, by Bézier curves [187]. Each element of the mesh through which the curve passes, is assigned material and thus forms the skeleton of the structure. The complexity of the

represented structure is determined by the number of control points. An additional thickness parameter defines the thickness of the connection, by adding layers of elements around the skeleton. Thus, the structure is defined by the location of the control points and the thickness values along the curves. A genetic algorithm and a constraint handling evolutionary algorithm have been applied for optimization [187, 188].

Wang and Tai encoded the Bézier curves in a graph, in which the vertices are the end- and control points and by applying a graph specific crossover [204]. Figure 3.5 shows an example of the Bézier curves representing a structure and the according graph. Several extensions have been proposed [186, 198, 200], for instance distinguishing between active and inactive curves that can change their state during the optimization [198], so that only active curves contribute to the structure. Furthermore, a hybrid genetic algorithm with local search for a multi-objective optimization has been proposed recently [201]. By a different research group [160, 163], a structure is defined more simply by connected piece-wise linear segments with different length and orientations.

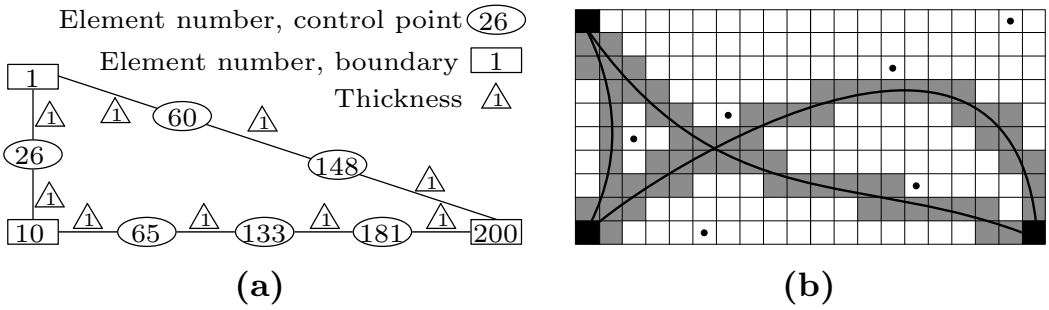


Figure 3.5: Graph representation that encodes control point locations and thickness values **(a)** that define Bézier curves surrounded by material in the design space **(b)** [204].

Bit-Array and Skeleton

Balamurugan et al. combined a Geometric Representation with a bit-array representation in a two phase approach [23]. In the first phase a bit-array encoding is applied and optimized by a two-stage adaptive genetic algorithm. They used a criterion for skeleton convergence as stopping criterion for the first phase. Originating from image processing "skeletonization" refers to the extraction of a thin skeleton structure from the bit-array solution. Connectivity and extent of the structure are preserved.

In the second phase the Geometric Representation is constructed from the optimized skeleton. Horizontally connected elements are aggregated to a rectangle with height of one element. The geometric variables are the left and right boundary elements of the rectangles. Rectangles are nodes in a graph and edges are defined from the connections of the converged skeleton to make the application of that graph specific cross-overs possible. A genetic algorithm is then used for optimization.

Complex Shaped Beam Elements

Sauter applied complex shaped beam elements as basic geometric units in a topology optimization [148, 149]. Start and end locations of the beams are the nodes of the graph and parameters corresponding to the edges define shape and thickness. He proposed straight, variable-thickness, and curved variable-thickness beams, realized by a cubic thickness distribution around a centre line, respectively. Furthermore, Sauter introduced specialized operators such as splitting and merging of beams. Due to the capability of this representation to change the displacement path via shape changes along the centre line, it is especially suited for the design of compliant mechanism and is applied to the design of an adaptive car seat concept.

Constructive Solid Geometry

Ahmed et al. propose a constructive solid geometry approach [2]. They combine primitives to obtain more complex geometries by using solid modelling operators such as the union operator. Two nodes and a thickness parameter define the rectangular beam primitives. A skeleton topology - i.e., the edges of the graph - are obtained by Delaunay triangulation. The coordinates of all nodes, other than fixed support and load nodes, are optimized and the triangulated mesh edges are decoded into a structure with rectangular beams with the corresponding thickness. For overlapping bars the constructive solid modelling union operator is applied in such a way that the complete area covered by the bars forms the structure. Figure 3.6 illustrates this idea. It is possible to meet a volume constraint by scaling the beam thickness values and a genetic algorithm is used for the optimization.

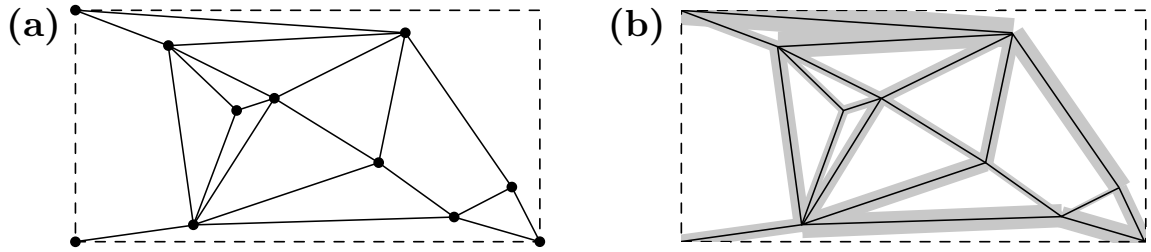


Figure 3.6: Representation by constructive solid geometry as in [2]: **(a)** nodes and triangulation, **(b)** beam shapes along edges.

3.3.4 Level Set Methods

In classical Level Set Methods [193], which were introduced in Sec. 2.1.3, the shape of the boundary between void and material is defined by the contours of a level set function. Usually, gradient-based optimization is best suited to address the large number of basis functions that compose the level set, but genetic algorithms have been used, as well [51]. Although the level set approach is not explicitly formulated, the Geometric Representations presented in Sec. 3.3.1, 3.3.2, 3.3.3 and 3.3.3, work on a similar principle. Some level set approaches are explicitly based on a smaller number of basis functions and can be optimized with evolutionary algorithms [71, 74]. Hybrid evolutionary strategies have been proposed to optimize a level set function based on beam-shaped geometric basis function [34, 35].

3.4 Indirect Representation

In Grid and Geometric Representations, a fixed descriptive model is used to directly describe the structure. In contrast to these cases, a representation can be classified as an "*Indirect Representation*", if the genotype encodes properties of a variable, generative model that *implicitly* defines material locations or a geometry. The rules or development processes that create the structure, are often inspired by onto- or morphogenesis processes in nature and mimic various aspects at different levels of abstraction.

3.4.1 Lindenmayer System

The grammar-based Lindenmayer system (L-system) is a model for embryonic cellular division of organisms. It was originally developed for modelling branched topology in plants. A planar graph is termed map

L system and defined as a finite set of regions, separated by edges that are connected at vertices. Biologically, the edges represent the cell boundaries. It has been used to evolve a family of three-dimensional table-like structures [85, 86]. Although no explicit reference is made to topology optimization and the structure is evaluated without a finite element analysis, this is effectively an early example for addressing a topology optimization problem with an Indirect (generative) Representation.

A more recent approach explicitly applies a map L-system to topology optimization [104, 138]. The model features cell division in two cells without the vanishing of cells. A first phase of the division applies markers to all edges. In a second phase, matching markers are connected, realizing the cell divisions. The two phases are repeated several times. Each division stage is followed by a dynamic stage until a force equilibrium is reached. Only the equilibrium result is used for the structural optimization. The development of a topology in a cellular division process is illustrated in Fig. 3.7. Additionally, a property that controls the thickness of the edges is introduced. The rules of the cellular division encoded in the genotype are optimized by a genetic algorithm.

The method has first been applied to aircraft structural design [104] followed by validation on a compliance cantilever [138]. The latter showed that the approach provides close to optimal solutions and that the number of evaluations is significantly lower compared to some grid-based [22, 205] and a Voronoi approach [73]. Further applications followed: aeroelastic topology optimization of flapping wing venation [176], simultaneous topology optimization of membrane wings and their compliant flapping mechanisms [177] and an extension for sub-system placement [119].

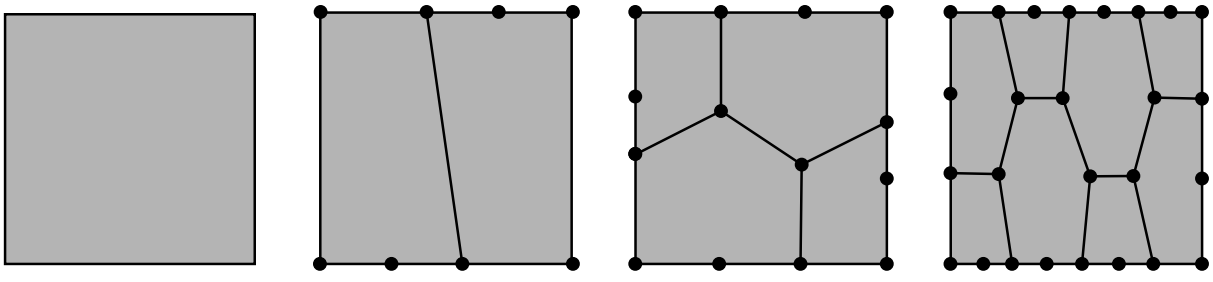


Figure 3.7: Starting point and the first three developmental steps of the cellular division process, shown at equilibrium [138].

3.4.2 Gene Regulatory Network

Steiner et.al. applied a biologically inspired cellular model for the evolutionary development of structures [33, 181, 182]. It features cell division and physical interaction of the cells, such as adhesion and repellence. The genotype encodes a simple form of a Gene Regulatory Network, which consists of a gene with a regulatory and a structural unit. The state of the regulatory unit is based on the cell type and the environment. It influences the structural unit, which determines the type of the cell and eventually controls a cell division. Steiner et al. also implemented a more complex cell model based on the same network structure, but with a higher number of units and including additional physical forces for adherence and repulsion and chemical gradients used for chemotaxis [183]. The chemical gradient causes an additional force that is acting on the cells, and can, also, be read by the regulatory units for gene activation.

The growing process and mapping to a three-dimensional structure is illustrated in Fig. 3.8. The development process is started from a single or few cells. Based on the encoded rules, the cells repeatedly divide and specialize in material and void cells and at the end of the development process, when the calculation area is filled, void cells are removed and material cells are mapped to the structure. An evolutionary multi-objective algorithm minimizes weight and inner stress [182] or bi-linear energy [183] within the elements of a three dimensional structure. The results show that the model is able to devise complex lightweight structures, especially in regions of the Pareto front of very low weight, in which classical gradient-based algorithms failed to find solutions.

3.4.3 Compositional Pattern Producing Network

A state-of-the-art method in the field of generative encoding is the Compositional Pattern Producing Network (CPPN) [179] as a model for a developmental process, often optimized by Neuro Evolution for Augmenting Topologies (NEAT), and hence called CPPN-NEAT. It is important to note that topology in this context refers to the topology of a neural network model. A CPPN is a model for embryonic development; more precisely it implements the specialization of a cell based on its position. The nodes encode various mathematical activation functions, that process the input coordinates of the cell. Using, for instance, Gaussian or periodic functions, desirable features, such as symmetry or repetition, can be realized. NEAT evolves a neural network model by starting from

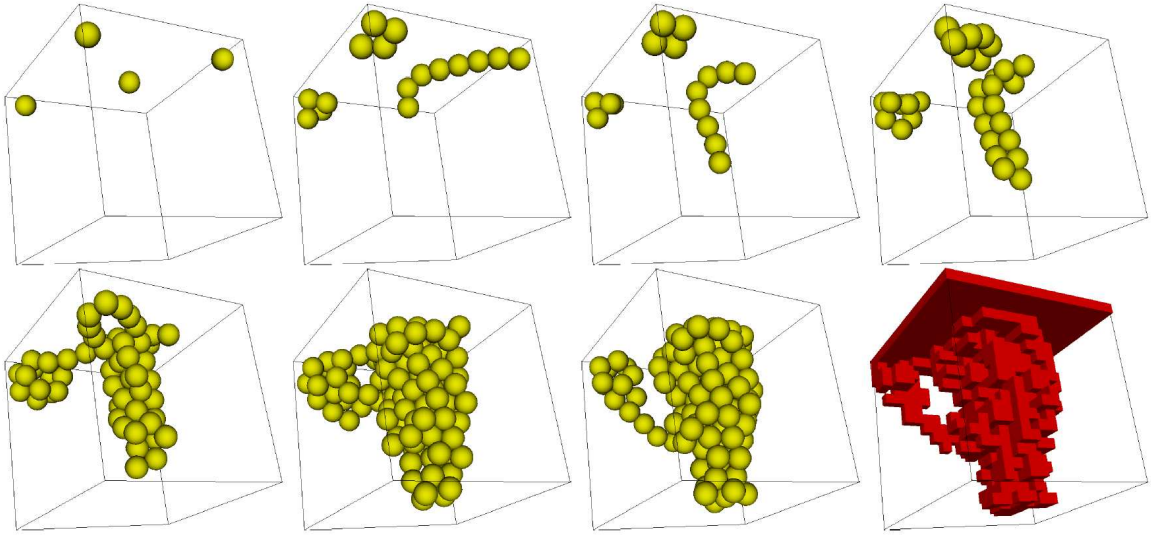


Figure 3.8: Cellular growth model based on motile polarized cells and voxelization, from Steiner et.al. [183].

a simple network and increasing its complexity.

Recently, CPPN-NEAT has been applied to topology optimization. Cheney et al. use the Cartesian and polar coordinates of the design space mesh as inputs to a CPPN [45]. The output of the network determines the material type and a multi-material structure is obtained. They define the objective function as the ability of the structure to match a frequency profile. Figure 3.9 depicts the representation with CPPN.

Evins et al. encode on the one hand the presence and absence of material and on the other hand the type of the material in two CPPN nets [62]. As inputs they use coordinates, as well and apply the method to optimize the ratio of generated energy to total cost of a photovoltaic collector.

3.5 Discussion

The previous sections introduced an overview and categorization of the existing representations that can be used for the problem of continuum topology optimization of mechanical structures with EC. Three categories of representations have been identified: Grid, Geometric and Indirect Representations. This section addresses the task of discussing the three classes and outlines the corresponding assets and challenges.

Grid Representation: A Grid Representation is relatively easy to realize, since it does not require the implementation of geometric bound-

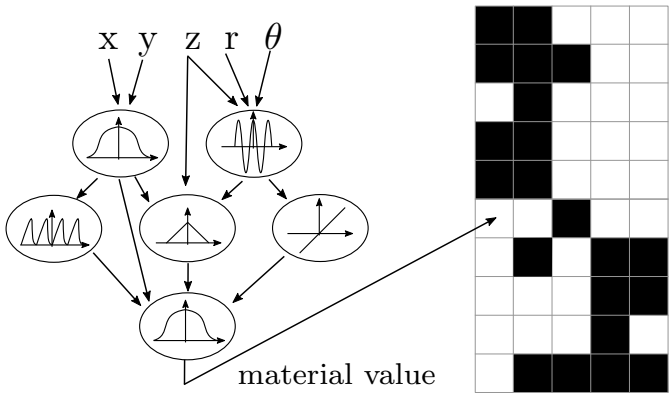


Figure 3.9: Example for CPPN representation, adapted from [45]. Cartesian and polar coordinates of the design space elements serve as input to a CPPN-NEAT model. The nodal activation functions are optimized and include for instance sine, Gaussian or sigmoidal functions. For each element the output of the network specifies the material within the corresponding element.

aries. The most common and simple mapping utilizes the computational grid of a finite element analysis model. This simplicity results in a reduced scalability, since the resolution of the discretization directly influences the number of variables. Therefore, Grid Representations are prohibitively expensive for fine grids, for which the search space dimensionality is higher and a lot of irrelevant solutions might be generated. Hence, variational operators have to be designed carefully, in order to reduce the amount of infeasible candidates; for instance, solutions that show numerical instabilities or a lack of connectedness. Yet, within the practical limit regarding the number of elements, it is always possible to represent the optimum phenotype can. In genotype-space the full design freedom is maintained and no assumptions on the solutions are implied. Thus, Grid Representations might have the highest potential for small-sized problems, for which little or no previous knowledge is available.

Geometric Representation: A Geometric Representation provides direct control over the potential complexity of a structural design, as it allows to choose the number of shape primitives. It decouples the mesh resolution from the dimensionality of the genotype by limiting the number of components within the structure. Prior knowledge about properties of good solutions is required; therefore, each Geometric Representation is most likely suited for a limited set of objective functions. Suitable shape primitives are good compromises between design flexibility and search space dimensionality and have the highest potential for this method.

Furthermore, Geometric Representations that facilitate the manufacturing of optimized structures from pre-defined components are beneficial for application. However, compared to Grid Representations, Geometric Representations involve a higher implementation effort, since they require a geometry mapping from the genotype to the finite element mesh. In conclusion, Geometric Representations might be most feasible for evolving structures with limited number of features, when valid assumptions regarding the choice of the shape primitives are available.

Indirect Representation: Optimizing a generative model might enable the exploitation of the potential of evolutionary computation, by creating and re-using problem-specific design patterns to evolve creative solutions. In comparison to Grid or Geometric Representations, few optimization variables can encode many structural details. However, finding and implementing a suitable Indirect Representation can be a challenging task and the simulation might be computationally expensive. Furthermore, it might not be clear, which phenotypes are in fact represented. Currently, the published literature on Indirect Representations is the least of the three categories and more research is required to determine which approaches are suitable for which kind of problems. For instance, indirect encoding using CPPN-NEAT in Sec. 3.4.3 was applied to complicated real-world examples, yet not evaluated on a standard compliance reference. Hence, it is not clear how accurate the optimum phenotype of a reference problem can be represented. Therefore, considering both sides, reference and application, might be an academically and practically rewarding direction, as indicated by the applications of the cellar division representation in Sec. 3.4.1.

So far, there has been little research interaction between the approaches from the field of evolutionary computation and the gradient-based topology optimization methods introduced in Chap. 2. Based on the work of Wu et al. [211] (which is an example of a Grid Representation), Sigmund [170] criticizes that non-gradient topology optimization methods can only be applied to problems with tiny mesh sizes, due to the exponentially growing computational cost. Advantages, claimed by Wu, include that the result does not contain intermediate density elements and that a global search is performed. Both claims are refuted as somewhat irrelevant, since finding a discrete solution for the global optimum on a tiny mesh is easily contested by finding a more precise solution with a more efficient gradient-based method on a fine mesh. A structural boundary that still contains

a thin layer of intermediate densities is usually present in density-based methods. However, this can be easily accounted for e.g. by applying a simple threshold to the intermediate densities of the result or by using more elaborated projection methods [169]. This leads to the conclusion that for typical topology optimization problems, for which gradient information is available, it is preferable to use it and non-gradient methods are not necessary.

With Geometric and Indirect Representations, there are interesting alternatives to Grid Representations, for which the number of parameters is decoupled from the number of mesh elements. These reduce the search space dimensionality by making assumptions about the solutions, such as geometric building blocks or the implicit generative process. However, these assumptions might compromise the ability to provide practically acceptable solutions, due to a restriction on possible phenotypes.

Yet, EAs and corresponding representations can also be applied to problems involving black-box simulations and are of advantage when applied to objective functions that involve non-linear characteristics such as disconnectedness, multi-modality, ruggedness, or noisiness.

In general, the application of methods from the field of computational intelligence was often met with scepticism. However, nowadays computational intelligence methods have been established for many applications. Examples for this development include EAs in the field of optimization or neural networks in the domain of statistical modelling, respectively machine intelligence. For the community researching in topology optimization this is yet a step to be taken.

Independently of the representation, most EC approaches for topology optimization are tested on objective functions of linear elastic problems. A majority thereof addresses various formulations of compliance objective functions subject to weight constraints or vice versa [2, 22, 23, 34, 42, 51, 94, 103, 110–112, 115, 129, 138, 152, 153, 155, 183, 187, 196, 205, 211], sometimes also formulated by means of displacement constraints [72, 73, 205]. Some authors have addressed compliant mechanisms [94, 148–153, 187, 199], inner stress [182], eigenvalue [115], stress and displacements under uncertainty [200] or target matching [186, 198, 201, 203] problems. Although the analytical sensitivities are known for these problems, they can serve as important references to evaluate the performance of an evolutionary optimization approach. (Therefore, the novel method proposed in this thesis is evaluated on the compliance problem in Chap. 5). Due to the efficiency and reliability of gradient-based methods, it seems unwise to claim superiority over gradient-based methods in these scenar-

ios; yet, it is still important to evaluate the optimality of the obtained structures and the number of evaluations.

Therefore, rather than in competition to gradient-based methods, in this work, EC approaches are seen as an alternative for especially difficult problems. However, this awareness demands that evolutionary approaches show their usefulness also in the face of more complex problems that are not commonly addressed by gradient-based methods. EC approaches for topology have been applied to a number of such problems: Topology optimization of piezo-electric sensor/actuator pairs for torsional vibration control [202], artificial magnetic meta materials [43], path generating compliant mechanisms [160, 162, 186, 188], compliant mechanism for the design of an adaptive car seat concept prototype [149], electrical motors subject to electromagnetic simulations [53, 54], flapping wing venation subject to aeroelastic analysis [176], simultaneous topology optimization of membrane wings and their compliant flapping mechanisms [177], photovoltaic collectors [62] and the design of random frequency profile structures [45].

In general, the application of derivative-free methods promises the highest impact when derivative information is "unavailable, unreliable or impractical to obtain"[143]. Practical engineering design optimization problems that are often involve black-box simulations are especially interesting. Although some examples have been presented, it is generally difficult to find literature that highlights unsolved problems, for which the stochastic topology optimization might be best suited - perhaps due to the nature of the publication process. Nevertheless, the following paragraphs complete the discussion by suggesting a few domains that could potentially benefit from the application of EC methods for topology optimization.

Interesting directions include problems that are challenging due to anisotropic materials (e.g. composites), coupled physics simulations (e.g. thermo-mechanical or fluid-structure), and simulations of manufacturing processes (e.g. metal-sheet forming or stamping processes); especially, when the simulations have to be addressed by explicit finite element solvers. In general, the inclusion of the manufacturing simulations or post-processing steps in the topology optimization design process might prove a valuable target for EC topology optimization methods. Guirguis et al. have done a first step by considering manufacturing and assembly cost of multi-component sheet metal structures in an one-stage approach [71].

Another important topic is crashworthiness topology optimization, which is introduced in Chap. 6 and which is the application focus of this thesis. Due to the strong non-linearities and transient nature of the models, analytical sensitivity information cannot be easily obtained

[66, 128]. Due to the simulation noise and the cost of the simulation, finite-differencing approaches are difficult to realize or infeasible, as well. Therefore, specialized heuristic methods are developed, which will be discussed in Sec. 6.1. However, these heuristics are not generic. Hence, they are prone to fail when outside of their specific use case. An example thereof it the engine hood topology optimization in [75], where actual head-injury criteria are addressed in a succeeding shape optimization, because it is not possible to explicitly address them in the topology optimization. Furthermore, EC method could be an alternative to existing heuristics for objective functions found in impact mechanics problems (with even more extreme dynamics and non-linearities than in a crash), e.g. in [69].

3.6 Summary of Contribution

This chapter provides a comprehensive review of representations that exist in the literature of evolutionary computation in the context of topology optimization. For the first time, the continuum topology optimization problem is discussed from the perspective of EC representations. Three categories, Grid, Geometric, and Indirect Representations are defined and the existing approaches are classified accordingly. This is followed by a holistic and critical discussion of the approaches, by outlining strengths and weaknesses, especially in comparison to gradient-based approaches. Although the majority of approaches in the explored literature applies Grid Representations, it is shown that interesting alternatives do exist, especially, in the field of indirect representations, which are only rarely researched at the moment. The presented review of the literature concludes with current and potential future fields of application and offers interesting directions for further research.

4 Topology Optimization by Predicting Sensitivities

As discussed in the previous chapter, evolutionary computation approaches for topology optimization come with the need of an appropriate lower-dimensional representation of the design space, since assigning a variable to each element is infeasible for anything but small-scaled problems. Furthermore, almost none of the existing evolutionary computation approaches make direct use of the detailed simulation data that contains valuable gradient information. These facts necessitate a novel topology optimization approach that is proposed in this chapter. It is introduced in Sec. 4.1. Its unique feature is a self-contained learning component that provides a generic search direction for improvement of the structure that systematically exploits structural state information. The approach utilizes a density-based formulation, but avoids a direct evolutionary optimization of a parameterized structure. The enclosing topology optimization algorithm is described in Sec. 4.2. The learning concept is either based on evolutionary optimization, proposed in Sec. 4.3 or supervised learning, proposed in Sec. 4.4. The hierarchy of the learning and modelling approaches presented in this chapter is shown in Fig. 4.1. Parts of this chapter are based on [9, 10, 14].

4.1 Generic Update-Signal Model

4.1.1 Introduction

A general evolutionary algorithm may perform well on the average of a large set of optimization problems, yet a method that incorporates domain knowledge usually has better performance when a specific type of problem is considered. Thus, it makes sense to tailor an evolutionary algorithm to a topology optimization problem by specific operators or by combining it with heuristics in order to increase its efficiency. Usually, the combination performs better than either of its components [59]. In fact, a

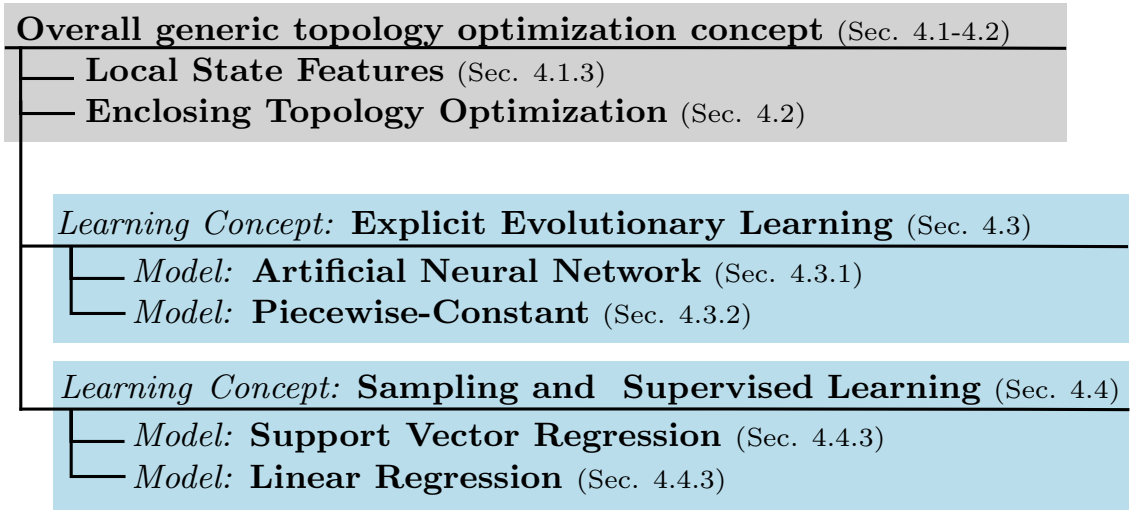


Figure 4.1: Overview of the structure of Chap. 4. The chapter introduces a generic topology optimization concept with different learning and modelling approaches.

number of hybrid approaches have been published that use sensitivity or stress related information to improve the performance of a global search for a topology optimization solution. For instance, Woon et al. used a mutation operator that used sensitivity contours to introduce cavities [209]. Others used elemental strain energy densities or stresses to bias the variation of the elements favourably [39, 44, 97, 100, 103, 107, 210]. Inversely, the generalization of an efficient, specialized method is also possible, i.e. to widen its applicability to be more general by reducing the amount of problem knowledge.

In this context, we dismiss the assumption that mathematical or numerical gradients are available. Concretely, the idea of the proposed method is to utilize a learning process that enables application of a classical topology optimization method even in cases, in which gradients are unavailable. Hence, the problem is changed from finding the optimum structure to finding the optimum search direction. Figure 4.2 illustrates this idea by depicting the flow of a conventional gradient-based topology optimization algorithm compared to that of a generic topology optimization algorithm.

In the classical approach in Fig. 4.2**(a)**, experts predefine mathematical expressions for sensitivities[1] *prior* to the execution of the optimization. The sensitivities depend on the problem and are rigorously derived depend-

[1] In topology optimization, typically, the partial derivative of a function with respect to a design variable is referred to as "*sensitivity*", compare Sec. 2.2.

ing on objective function and constraints. The state information, that is required for sensitivity computation at the given design point is obtained from one or several structural analyses. The resulting method works for a specific problem only. For any modifications, other corresponding derivative formulations need to be found externally to the optimization process. One one hand, the advantage of this process is that an accurate search direction is provided. On the other hand, the problem here is obviously that the capability to find a solution depends on extrinsic expert knowledge.

In contrast, the proposed, novel approach in Fig. 4.2**(b)** aims instead at providing a generic model for the sensitivity in case no expert knowledge is available. The model is generated as an intrinsic part of the optimization process. For the purpose of defining a search direction, state data is systematically mapped onto an heuristic substitute for a sensitivity formulation. This substitute model may provide an approximation of the gradient, yet it may also apply any other function that achieves improvement of the structure. Therefore, throughout this work it is generally designated as "update-signal model". Two possibilities for obtaining the update-signals are presented later in this chapter and are indicated in the figure: evolutionary optimization and supervised learning.

It should be clear that the generic optimization performs similarly to the gradient-based one, in case the update-signal model accurately predicts the correct sensitivities; therefore, the proposed approach can be termed "Topology Optimization by Predicting Sensitivities"(TOPS). Such an approach has not been implemented yet in the known literature.

4.1.2 Replacing Sensitivities

Due to their high efficiency and generality, density-based methods are one of the most commonly applied topology optimization approaches in both industry and academia. The majority of topology optimization problems can be formulated as the general, discretized optimization problem with volume constraint from (2.4). For reading convenience it is repeated here:

$$\begin{aligned} &\min_{\boldsymbol{\rho}} \; F(\boldsymbol{\rho}, \mathbf{u}(\boldsymbol{\rho})) \\ &\text{s.t.} : G_0(\boldsymbol{\rho}) = V(\boldsymbol{\rho}) - V_0 \leq 0, \\ &\qquad 0 \leq \rho_{\min} \leq \rho_i \leq 1, \; i = 1, \ldots, N \;\; , \end{aligned} \tag{4.1}$$

where the design variables $\rho_i,\ i = 1, \ldots, N$ are the material densities within finite element i and material parameters are interpolated based on the

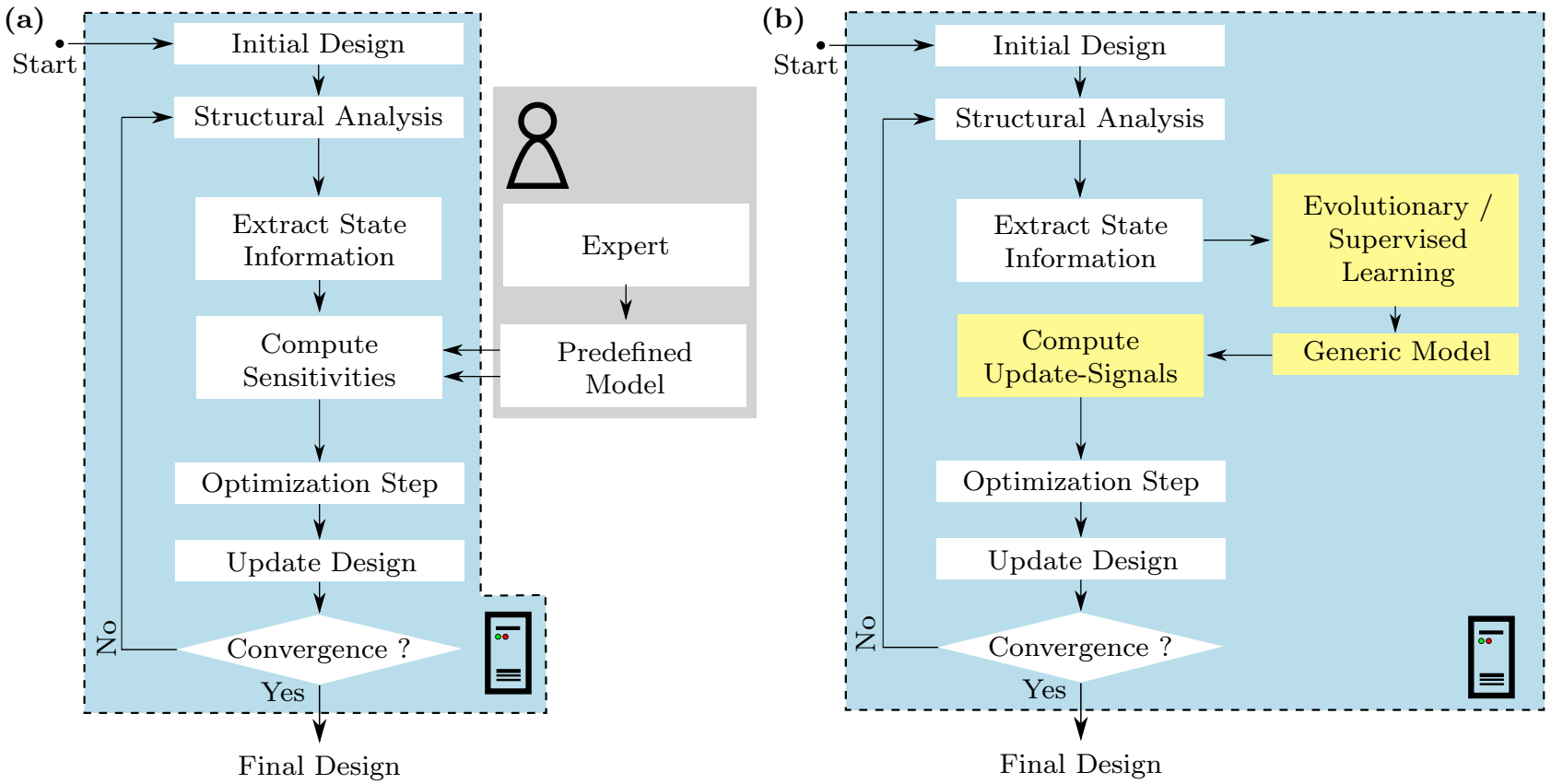


Figure 4.2: The concept of standard gradient-based topology optimization **(a)** compared to the concept of generic topology optimization **(b)**. In the generic flow, the sensitivity modelling is performed by a computational intelligence approach instead of a human.

SIMP scheme in (2.3). Since the approach introduced in this chapter focuses solely on an objective function with volume constraint, the additional constraints are dropped; yet, the following considerations could also be applied to additional constraints. Associated with problem (4.1) is a structural analysis (typically a finite element analysis) that yields the structural state $\mathbf{u}$.

The efficiency of density-based methods with gradients comes from the fact that the sensitivities of all variables with respect to the objective function and constraints can be obtained by at most one finite element analysis per objective and constraint. This methodology is very efficient in topology optimization, where the number of variables can reach up to millions and the number of constraints is usually moderate. One has to note that it is usually necessary to conduct an additional adjoint analysis for every constraint plus the objective function [1, 6, 30, 68]. Examples for adjoint analysis for compliance and displacement minimizations are given in App. A.1.

To address problem (4.1) with gradient-based methods, the partial derivative, or sensitivity, of F with respect to the variable ρ_i is required. In the context of this work, emphasize has to be put on the global state vector $\mathbf{u}$, which is required to compute the objective value. Beyond that, gradient-based approaches also utilize this state vector to compute the sensitivities of the objective function, with respect to the design variables.

When topology optimization problems are analyzed based on adjoint sensitivity analysis [1, 6, 30, 68], the mathematical sensitivity formulation can be expressed based on the state $\mathbf{u}$ and, depending on the problem, adjoint states $\mathbf{v}_1$, $\mathbf{v}_2$, ... :

$$\frac{\partial F(\boldsymbol{\rho})}{\partial \rho_i} = \frac{\partial F(\boldsymbol{\rho})}{\partial \rho_i}\left(\boldsymbol{\rho}, \mathbf{u}, \mathbf{v}_1, \mathbf{v}_2 \ \dots .\right) \ ,$$

In density-based topology optimization all variables represent the same structural property, i.e. the material density within an element of the design space. The benefit of this precondition is that the sensitivity expression can be *localized* in common cases the such as compliance minimization, minimizing heat conduction resistance, or compliant mechanism synthesis [30]. When a closed mathematical expression for the localized sensitivity is found, this implies that only *local* state information is required, to compute the sensitivity of variable i i.e.

$$\frac{\partial F(\boldsymbol{\rho})}{\partial \rho_i} = \frac{\partial F(\boldsymbol{\rho})}{\partial \rho_i}\left(\boldsymbol{\rho}, \mathbf{u}_i, \mathbf{v}_{1_i}, \mathbf{v}_{2_i} \ \dots\right) \ , \tag{4.2}$$

with $\mathbf{u}_i$, $\mathbf{v}_{1_i}, \mathbf{v}_{2_i}$ being the localized versions of the states. The index denotes state information that is associated with element i and the corresponding nodes in the mesh. The localization effect is caused by the partial derivative of structural characteristics. For instance, the derivative of the stiffness matrix is only non-zero for the components influenced by the element that corresponds to the variable. Accordingly, when dealing with structural optimization, the local elemental state(s) contains valuable information on how beneficial with respect to the objective function or constraints, it is to increase or decrease a design variable. It is important to realize that the local state information implicitly contains global information from the structural analysis, since it depends on the design: $\mathbf{u}_i = \mathbf{u}_i(\boldsymbol{\rho})$.

Hence, a mapping that updates the design variables, i.e. redistributes material within the structure, can be defined based on the localized states. For one objective function without additional constraints as in (4.1), the sensitivities (4.2) can be used to update the solution:

$$\mathbf{update}_{\mathbb{R}^N \to \mathbb{R}^N} : \\ \boldsymbol{\rho}^{(k)} \to \boldsymbol{\rho}^{(k+1)} \left(\left[\frac{\partial F^{(k)}(\boldsymbol{\rho})}{\partial \rho_1} \;\; \dots \;\; \frac{\partial F^{(k)}(\boldsymbol{\rho})}{\partial \rho_i} \;\; \dots \;\; \frac{\partial F^{(k)}(\boldsymbol{\rho})}{\partial \rho_N} \right]^T \right) \;\; ,$$

in which the mapping **update** is an optimization step, which provides a new vector of design variables $\boldsymbol{\rho}^{(k+1)}$ based on the gradient, and k is the iteration number.

In a generic optimization, the goal is to perform a comparable optimization step, based on update-signals that substitute the sensitivities:

$$\mathbf{update}_{\mathbb{R}^N \to \mathbb{R}^N} : \boldsymbol{\rho}^{(k)} \to \boldsymbol{\rho}^{(k+1)} \left(\left[S^{(k)}_{\boldsymbol{\theta}1} \;\; \dots \;\; S^{(k)}_{\boldsymbol{\theta}i} \;\; \dots \;\; S^{(k)}_{\boldsymbol{\theta}N} \right]^T \right) \;\; , \tag{4.3}$$

where $S_{\boldsymbol{\theta}}$ is a model with parameters $\boldsymbol{\theta} \in \mathbb{R}^{\Theta}$, that provides the update-signals $S_{\boldsymbol{\theta}i}$ for all elements.

Under the auspices of this approach, it makes sense to begin by targeting an update-signal model that predicts sensitivities. Ideally, one would use the same localized state vectors that are required by the sensitivities, contained in independent variables for the model input:

$$S_{\boldsymbol{\theta}} = S_{\boldsymbol{\theta}} \left(\boldsymbol{\rho}, \mathbf{u}_i, \mathbf{v}_{1_i}, \mathbf{v}_{2_i}, \dots \right) \sim \frac{\partial F(\boldsymbol{\rho})}{\partial \rho_i} \;\; . \tag{4.4}$$

By processing the corresponding local simulation data, a *single* model can provide update-signals for all variables.

For the determination of the update-signal model Sec. 4.3 and Sec. 4.4 will consider evolutionary optimization and supervised learning methods, respectively. The learning tasks are limited to determine Θ model parameters instead of N density values, where it is possible to chose the model so that $\Theta \ll N$. The *generic* TOPS method with the novel learning component has the following conceptual advantages compared to conventional EC approaches from Chap. 3:

- The update-signal model is taking the place of gradients. Hence, the task is to determine a model for the search direction, which makes it possible and consequent to utilize well-established topology optimization techniques. This in turn enables the application of gradient-based algorithms, even when no mathematical gradient on the design variables is available. Existing optimizers, regularization methods, and interpolation schemes for topology optimization can be utilized out of the box.

- TOPS maintains the full design flexibility, since each element of the finite element mesh of the design space is assigned an individual design variable. Since the update-signal model is decoupled from the design space mesh, it does not need to change with a higher mesh resolution. Moreover, it does not reduce the search space dimensionality of the design space by making assumptions about the solutions that jeopardize the ability of expressing the optimal structure.

- The method makes systematic use of state information that is obtained from the simulation data of structural analyses. Since the state is the source for obtaining gradients, it can provide search directions for updates of the structure. When available, this information should not be neglected in favour of a purely stochastic search. Also, previous analytical or semi-analytical knowledge can be utilized. In theory, once a sufficiently accurate and general update-signal model is found, the efficiency of a gradient-based search is approached.

4.1.3 Local State Features

The concept of modelling characteristics of the optimization problem is not new and, for instance, used in surrogate-assisted optimization for engineering [65, 96]: response surface methods generally [123], in the context of structural optimization [156], or as well in estimation of distribution approaches [80, 106]. However, these approaches directly model the objective

function, or the distribution of favourable solutions in the space of *design variables*. Yet, in (4.4), a different approach is expressed by the dependency on the structural *state(s)*. This approach expresses the exploitation of localized state information, associated with variables as model inputs. These inputs lead to modelling a search direction in the space defined by the localized state features, in contrast to modelling the objective function. This difference is essential, since the high number of variables in topology optimization renders the learning of distributions or surrogates of the objective function practically infeasible.

Besides the mathematical perspective on sensitivity, the idea of a generic update-signal is motivated by gradient-like, biologically-inspired (but non EC) approaches in the literature. State information is sometimes used in an intuitive way instead of computing rigorous mathematical sensitivities. Heuristics are gained from human intuition, experiments and/or biological motivation. These approaches utilize state information for heuristically conducting topology optimization very similar to gradient-based topology optimization,

An early representative of these methods is the so-called "soft kill option", inspired by the natural growth processes in trees, devised by Baumgartner and Mattheck [25, 116]. The topology optimization is performed similar to density methods: Based on the elemental stress, the Young's modulus of the material in the element is reduced, slowly weakening the unnecessary elements in the mesh and reinforcing the more stressed parts. This yields structures similar to those of minimum compliance optimization results. Similarly, early ESO/SERA approaches, (see Sec. 2.1.3), were mainly based on intuitively defined heuristic criteria, e.g. they used the elemental stress as rejection criterion to remove whole material elements from the design space [212, 213]. Tovar [190, 191] showed that the bone-remodelling process in human trabecular structures can be modelled by a topology optimization process that achieves a uniform distribution of the strain energy density throughout the structure. Bone material is accumulated where the strain energy is large and reduced where strain energy is low. The process of bone-remodelling in a fixed tissue region is modelled as a Hybrid Cellular Automaton (HCA), in which each cell holds a continuous state that represents the amount of material in the element. Each cell updates its state based on the local strain energy densities, including those of neighbouring cells. Patel applied the HCA algorithm as an heuristic topology optimization method in the field of crashworthiness problems [131, 132]. The method achieves an optimized result by using elemental internal energy densities occurring during the crash event (hence

state information) for the update of the cells.

These heuristic approaches have a great similarity to the classical, gradient-based density approaches. For instance, as stated by Sigmund [171] only a few lines of code need to be changed to turn the density-based topology optimization code [168] into a cellular automaton algorithm. In both cases elements or cells can develop into material or void. Hence, the classical mathematically rigorous approaches as well as the mentioned gradient-like, heuristic approaches fall into the category of Fig. 4.2**(a)**, since the rule for material redistribution is defined by a user prior to the optimization. In both cases, the resulting concept is only applicable for the optimization of a certain objective function.

The target of the generic approach proposed in Fig. 4.2**(b)** is to exploit localized state information systematically and to obtain an update-signal suitable for the optimization objective. This should be achieved without need of the human expertise of devising a predefined heuristic or sensitivity analysis. Hence, the state information serves as input features in an autonomous model learning process. Since in (4.2) localized state information is assumed, we therefore term the inputs of the update-signal model Local State Features (LSF).

Then, with a generalized LSF vector $\mathbf{s}_i \in \mathbb{R}^J$, we can express the update-signal model as:

$$S_{\boldsymbol{\theta} i} = S_{\boldsymbol{\theta}} \left(\boldsymbol{\rho}, \mathbf{u}_i, \mathbf{v}_{1_i}, \mathbf{v}_{2_i}, \dots \right) = S_{\boldsymbol{\theta}} \left(\mathbf{s}_i \right) \quad ,$$

where J is the number of LSF. Note that the feature vector just as the states depends on the current design: $\mathbf{s}_i = \mathbf{s}_i(\boldsymbol{\rho})$.

The actual LSFs in the components of $\mathbf{s}_i$ that are applied depend on the problem. Depending on the objective function and the type of analysis, different LSF may be available. A trivial LSF is the amount of material density in an element ρ_i i.e. the design variable associated with the element. When the governing equation is solved by a (displacement-based) finite element solver, the state vector is the vector of nodal displacements $\mathbf{u}$. The most basic LSF are naturally the components of the elemental displacement vector $\mathbf{u}_i$.

As an example, Fig. 4.3 presents the state of a two-dimensional, square finite element, which is defined by four nodes. The element is shown in undeformed and a deformed state. The deformed state is the result of applying a load and performing the finite element analysis. The state is "local" with respect to the element considered. For the displayed, two-dimensional square element i, an eight-dimensional local displacement vector $\mathbf{u_i} = [u_{i1}, \dots, u_{i8}]^{\mathrm{T}}$ forms the basic LSFs.

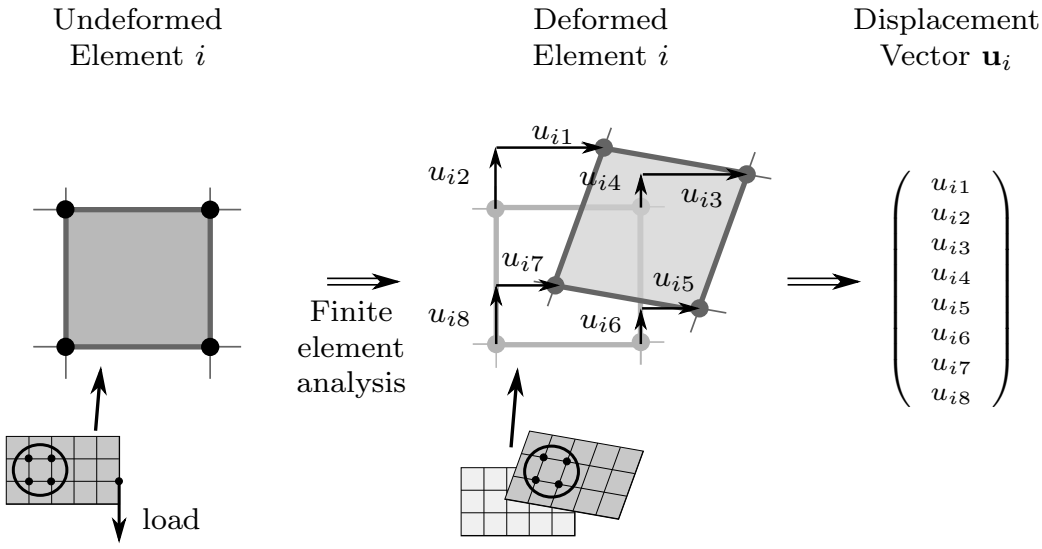


Figure 4.3: Local state features: Example for displacements of element nodes when a finite element analysis is performed.

A simple scenario, in which this information is utilized is the minimum compliance problem. Here, the local, elemental strain energy density, which depends on the local displacements, indicates whether material should be added or removed in the corresponding spot. When this relation is implemented as an iterative rule, a compliance minimization is performed.

Figure 4.4 illustrates LSFs of a two-dimensional cantilever structure subject to a load (as considered in the fundamentals, Sec. 2.5). Different (normalized) features are highlighted. Elements are colour coded with the first and second component of the local displacement vector and the elemental strain energy. It can be seen that the displacements of the first node contain less structured information, compared to the elemental strain energy.

Computing elemental energies, strains, or stresses, highlights that it is possible to post-process basic LSF, such as nodal displacements, to obtain stronger features, which represent more concise information. This processing also provides an opportunity to incorporate optional expert knowledge. Intuitive, heuristic, semi-analytical or analytical information on the sensitivities can be included, depending on the problem. For any learning process that aims to map the LSF onto update-signals, the information contained in the LSFs is of critical importance. On the one hand, for weak LSF, which contain little or very abstract sensitivity information, the learning task can be challenging. On the other hand, with

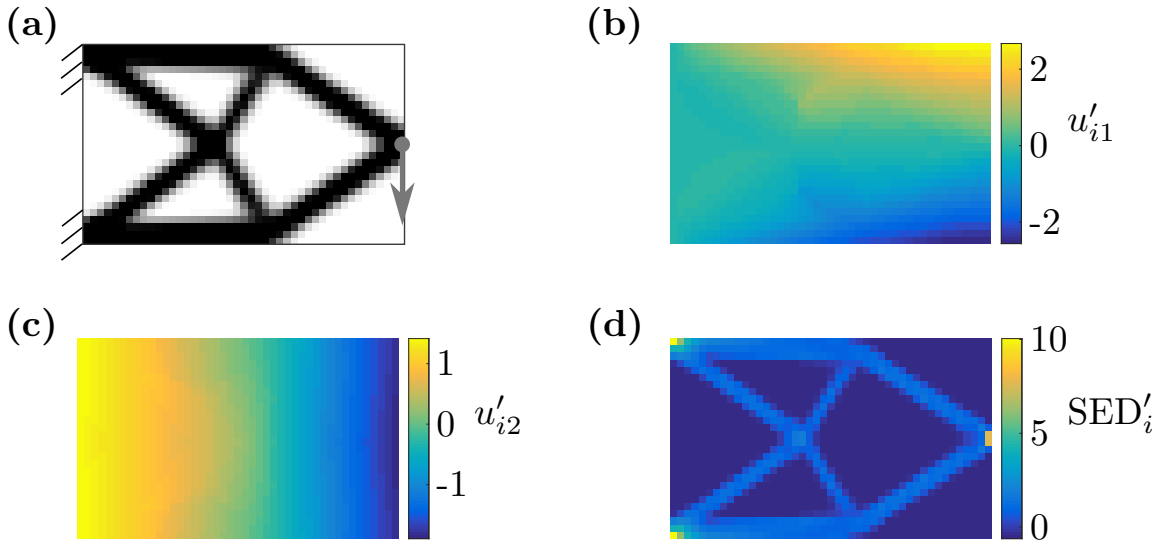


Figure 4.4: Example for: **(a)** Material distribution of cantilever structure subject to static load in the design space (note that white area contains minimum density material); Resulting Local State Features: **(b)** horizontal displacement of first element node u'_{i1}, **(c)** vertical displacement of first element node u'_{i2}, **(d)** elemental strain energy SED'_i. The stroke indicates that the LSFs are normalized over the design space to zero mean and standard deviation one, hence, unitless values.

increasing quality of LSFs, the learning task is simplified and the usage of mathematical sensitivities is approximated.

Since the discussed LSF are continuous (physical displacements, densities), the task is to map a continuous input parameter space to a continuous output parameter, hence a regression model is required. Two machine learning methods for implementing the update-signal model are proposed in Sec. 4.3 and 4.4. Before that, optimizer and regularization methods, as well as the computational flow of TOPS are described in the next section.

4.2 Enclosing Topology Optimization

So far, it has not been stated how the update-signal is concretely used. In the TOPS approach, the concept of the generic update-signal is enclosed in techniques from density-based topology optimization.

An efficient scheme to obtain the components $\rho_i^{(k+1)}$ of an updated so-

lution $\boldsymbol{\rho}_i^{(k+1)}$ is the Optimality Criteria (OC) update [30]:

$$\rho_i^{(k+1)} = \begin{cases} \max(\rho_{\min},\, \rho_i^{(k)} - m) \\ \quad \text{if } \rho_i^{(k)} B_i^{\eta} \leq \max(\rho_{\min}, \rho_i^{(k)} - m), \\ \\ \min(1, \rho_i^{(k)} + m) \\ \quad \text{if } \min(1,\, \rho_i^{(k)} + m) \leq \rho_i^{(k)} B_i^{\eta}, \\ \\ \rho_i^{(k)} B_i^{\eta} \text{ else } , \end{cases} \tag{4.5}$$

in which k is iteration number, m is a move limit, and η is a damping parameter. These are stability parameters that control the amount of change that occurs during any one iteration of the topology optimization. A recommended value for the damping parameter is $\eta = 0.5$.

In the proposed generic approach, the OC-input

$$B = \max(0, -\widehat{S}_{\boldsymbol{\theta}}(\mathbf{s}_i))/\Lambda$$

depends on a filtered version of the update-signal $\widehat{S}_{\boldsymbol{\theta}}(\mathbf{s}_i)$ and a Lagrange multiplier Λ. The formulation assures that $B \geq 0$ holds. The update-signal is treated exactly as mathematically derived optimality criteria would be. The smaller the error

$$\left|\frac{\partial f}{\partial \rho_i} - S_{\boldsymbol{\theta}}(\mathbf{s}_i)\right| \ ,$$

the more the optimization is expected to yield the results of a gradient-based optimization.

The OC-update is an intuitive, heuristic scheme that increases the density in elements with low sensitivities and decreases the density in elements with high sensitivity (assuming minimization). It does so, while maintaining the volume constraint by adjusting the Lagrange multiplier Λ with a bi-sectioning algorithm. The OC-update is the concrete implementation of the **update** mapping in (4.3) that is used throughout this work.

The hat indicates the filtered version of the update-signal, according to a sensitivity filtering approach [167]. Concretely, the update-signal of an element is modified according to:

$$\widehat{S}_{\boldsymbol{\theta}}(\mathbf{s}_i) = \frac{1}{\rho_i \sum_{j=1}^{N} \widehat{H}_{ij}} \sum_{j=1}^{N} \widehat{H}_{ij} \rho_j S_{\boldsymbol{\theta}}(\mathbf{s}_j) \tag{4.6}$$

and

$$\widehat{H}_{ij} = \begin{cases} r_{\min} - \mathrm{dist}(i,j) & \mathrm{dist}(i,j) < r_{\min} \\ 0 & \text{otherwise} \end{cases} ,$$

where $\mathrm{dist}(i,j)$ is the centre to centre distance of elements i and j. Essentially, the update-signal of an element is replaced by the weighted average of the update-signals of its neighbours, the elements within radius $r_{\min} > 0$. The filtering is a method that imposes a minimum length scale on the resulting structures and helps to avoid numerical instabilities as discussed in Sec. 2.2.2.

These essential components, optimizer and filter, are chosen as in state-of-the-art density-based topology optimization algorithms [5], but they are adapted to be used in the TOPS approach. Although the OC optimizer and filtering of sensitivities are well established and efficient techniques that are suitable for numerous problems, these components can be exchanged, for example in order to address a problem with extra constraints as in (2.4) with mathematical programming methods.

The computational flow of this basic TOPS variant can be seen in Fig. 4.5**(a)**. According to the flow, the topology optimization is started with the evaluation of the initial solution. Then, a first training of the update-signal model is performed. In the subsequent step, the trained model provides the update-signal. The signal is filtered according to (4.6) and a new structure is obtained by applying the OC-update in (4.5). The new structure is evaluated and it is checked whether a stopping criterion is fulfilled. If this is not the case a new model is trained for the new structure.

The next section modifies the algorithm to conduct the training only when it is actually necessary.

4.2.1 Improvement Threshold

This section introduces an *improvement threshold* as extension to the basic TOPS variant, which was presented in the preceding section. The target is to reduce the duration and total number of model learning events, since training of the model necessarily involves expensive function evaluations. The idea behind this is to reduce the learning of the model to cases, when it is actually necessary in order to achieve improvement of the structure. This will be implemented by adding a check of the actual progress and a consequent branch into the computation scheme.

In the basic TOPS approach in Fig. 4.5**(a)**, the design is updated and

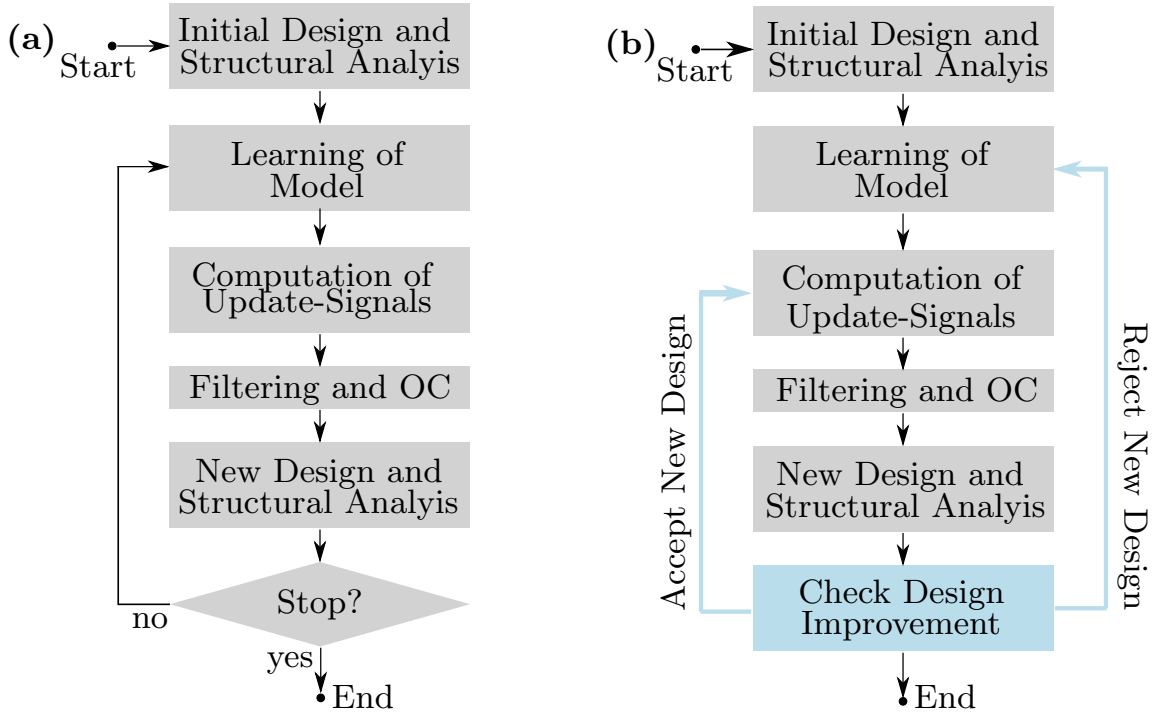


Figure 4.5: Computational Flow of TOPS **(a)** and TOPS with improvement threshold **(b)**.

accepted without further restrictions when the training is completed. A new model is learned every iteration. This neglects potential generalization of the model. The update-signal model may be reused for several subsequent updates of the design without re-training.

The difficulty here lies in determining the point at which a model is sufficiently accurate to stop the training or, respectively, skip it completely. This problem is resolved by an improvement threshold, i.e. a minimum progress that has to be achieved by the structural update. The corresponding, modified computational flow of TOPS is shown in Fig. 4.5**(b)**. The modification of TOPS is visible by the possibility to reject designs in a "Check Design Improvement"-step and using the rejection as trigger for the training.

Under the assumption of minimization, the progress of the topology optimization can be quantified by the relative design improvement ΔF, with

$$\Delta F = \left| \left(F^{(k)} - F^{(k+1)} \right) / F^{(k)} \right| \quad , \tag{4.7}$$

where k is the iteration number, $F^{(k)}$ is the value of the objective function and we assume $F^{(k)} > F^{(k+1)}$. The optimization is stopped when the

design improvement becomes smaller than a defined threshold: $\Delta F < \Delta F_{\min}$.

A simple stopping criteria for training of the model parameters is to reach a fixed number of function evaluations. This is also motivated by the fact that in industrial applications often a budget of allowed evaluations exists. The computational flow of the improvement threshold augmentation is shown in Fig. 4.6. It corresponds to a detailed description of the check design improvement step in Fig. 4.5**(b)**.

We define a number of evaluations M, after which the design improvement (4.7) is checked. Further, we limit the evaluations during one topology optimization iteration by $M^{\max}$. Following the flow in Fig. 4.5**(b)**, the training is started in the first iteration and runs for M evaluations. After M evaluations it is interrupted, the design is updated and the design improvement is checked against the threshold. The relative design improvement ΔF is measured as defined in (4.7). A counter variable $M_{\text{iter}} = 0$ is initialized.

1. **If** $\Delta F >= \Delta F_{\min}$: The design is improved sufficiently and thus the model training is completed for this iteration. The design is accepted, the topology optimization continues and M_{iter} is set to zero. In the next iteration, the model is reused for the update of the design. **Otherwise**: M_{iter} is increased by M and it is continued at 2.

2. **If** $M_{\text{iter}} < M^{\max}$: The model training is continued for another M evaluations. After training is completed the updated design is checked again by 1. **Otherwise**: The overall optimization process is

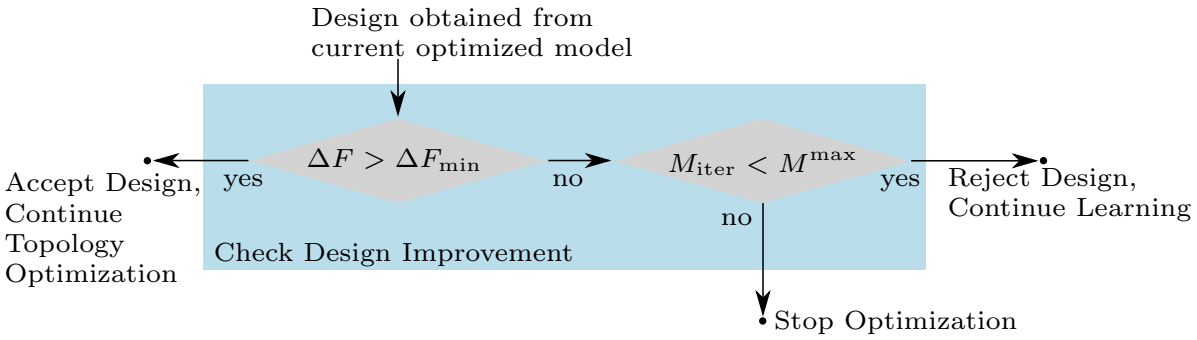


Figure 4.6: The "Check Design Improvement" Step of TOPS with Improvement Threshold.

stopped and the design is rejected. The preceding design is the final result of the optimization.

This scheme is motivated by the fact that the required effort for model learning is unknown. In step 1, a fixed number of the evaluations is performed to train the model. The algorithm updates the design iteratively, and, in case of failure, the training step is continued, so that only the minimum amount of evaluations for acceptance are required. However, when the design is not improving, this can either be caused by its convergence or by insufficient quality of the update-signal model. Therefore, a limit on the maximum amount of evaluations is provided. Furthermore, it may happen that weak predictions still result in a marginal improvement of the design. In order to avoid this case $\Delta F_{\min} > 0$ is specified, enforcing a minimum improvement threshold of the structure and, implicitly, a minimum for the quality of the model[2].

Assuming that the design is not yet converged and M and $M^{\max}$ are set sufficiently large, this scheme trains the model until it eventually achieves the specified improvement $\Delta F_{\min}$. It avoids an early-stopping behaviour that might be caused by low quality models that would occur when $\Delta F < \Delta F_{\min}$. Furthermore, the number of evaluations can be reduced, since the model is reused as long as the structure is improving.

4.3 Explicit Evolutionary Learning

Significantly, none of the existing EC approaches for topology optimization exploits the potential of LSFs. This is not too surprising, if we consider that these are naturally black-box approaches, which are characterized by their universality. Domain-specific knowledge is more commonly incorporated by designing adequate representations or variational operators for a problem. Only in recent works [45, 62] described in Sec. 3.4.3, coordinate information is utilized. The structure is represented by a mapping of the local coordinates, such as Cartesian or polar coordinates to a binary value describing presence or absence of material at that location. Yet coordinates are not state information; on the contrary, they are independent of the structural analysis, and, therefore, independent of the current structure.

[2] An adaptive improvement threshold that aims at reducing the sensitivity of the optimization to the choice of parameters, may be used, as well. This was applied in a publication of the author [15], however, for the sake of brevity, results are not included in this work.

In the proposed TOPS method, evolutionary optimization can be applied to tune the parameters of the update-signal model, resulting in an indirect and adaptive representation. The effect of the model on the structure can be observed by its application to modify the structural design. Model parameters, which result in an adequate update-signal, will achieve (until convergence) an improvement of the structure. These parameters can be optimized explicitly by an evolutionary search algorithm using the objective value of the structure as optimization target. Fig. 4.7 illustrates the optimization idea of a population of models that determine an update of the structure.

Algorithms that combine a global search with a local search are often called memetic algorithms [124]. Conventionally, a local search is applied to enhance the global search. However, the proposed method instead uses the global optimizer in order to provide a direction in a local search. From the EC perspective this is an implicit representation. It is implicit because a mapping from the state of the structure to an update of the design, i.e. a rule for the changing of the design is represented and optimized.

Consequently, the model parameters are directly encoded sequentially as optimization variables. In order to devise a suitable update-signal model, $S_{\boldsymbol{\theta}}(\mathbf{s}_i)$ the parameters $\boldsymbol{\theta}$ are optimized by the evolutionary optimization, i.e. the objective value of the design after the update step is minimized. Formally, the objective value of a candidate $\boldsymbol{\theta}$ is evaluated by emulating an update of the design:

$$\min_{\boldsymbol{\theta}} F(\boldsymbol{\rho}^{(\mathrm{k}+1)}(\boldsymbol{\theta})) \ . \tag{4.8}$$

This emulation includes the computation of the update-signals for all design variables, the filtering and the OC-update to obtain $\boldsymbol{\rho}^{(\mathrm{k}+1)}$, and computation of the objective value of the new resulting design vector $F(\boldsymbol{\rho}^{(\mathrm{k}+1)})$ by a structural analysis.

Expression (4.4) suggests that the model targets to predict sensitivities of the design variables, which should in many cases lead to improvement of the structure. However, when objective functions are significantly non-linear, it might be difficult to obtain satisfying results with gradient-descent. If the linearity assumption or the gradient are incorrect, the structure will not improve as expected. In this case, a model that is learned by evolutionary optimization will serve as an indirect representation that provides a rule for updating the design. Hence, even when the most accurate gradient does not improve the structure, the evolutionary optimization might still be able to provide an alternative direction that

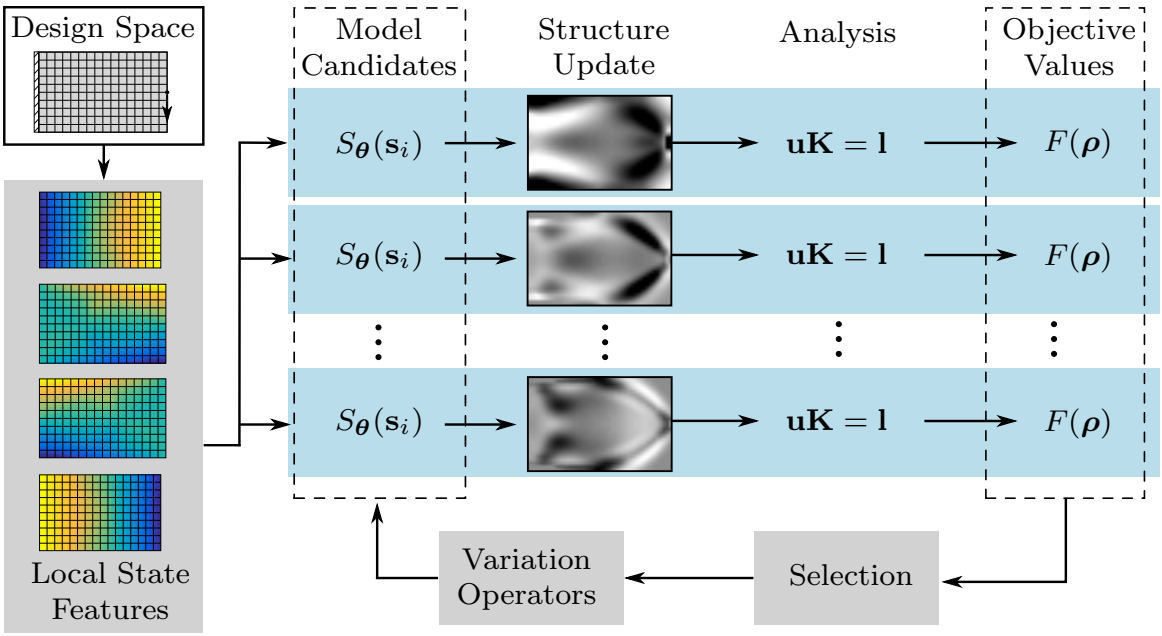


Figure 4.7: The optimization process of updating the structure based on local state features and model.

improves it anyway.

In the following, two concrete update-signal models are presented. The first model is based on the machine learning technique of neuro-evolution and the second one is a piecewise-constant model based on the idea of a state based representation.

4.3.1 Neural Network Approximation Model

Artificial neural network approximation models are biologically-inspired statistical models from the field of computational intelligence. For background information, the reader is referred to standard literature [32].

In the field of computational intelligence, Neuro-Evolution(NE) is the combination of evolutionary algorithms with neural networks [215] and is an active area of research. An overview on NE can be found in [63]. It considers the application of evolutionary algorithms to devise efficient neural networks for various tasks in a reinforcement learning-like optimization process. Neuro-evolution has been applied, for instance, to controller design for pole-balancing [82, 93, 178], controller design for wing shape adaptation with concurrent sensor and actuator evolution [172], controllers for multi-legged robots [192], classification tasks [40], and computer games [142, 180]. Instead of manually improving the neural network model, NE

explores networks by evolutionary optimization. Besides from practical application this may also be used to mimic and improve understanding of evolution of biological neural systems.

Since the ideal configuration is unknown, more elaborate approaches apply changes of the network topology or activation functions. Therefore, such modifications are difficult to perform, whether manually or by using standard gradient-based learning approaches. In approaches such as Neuro-Evolution by Augmenting Topologies [178], the complete network architecture is evolved, starting from a single neuron. In the simplest case of NE (which is considered here), only the parameters within a fixed neural network are optimized.

Since the goal is to provide a generic topology optimization method a fairly general model should be chosen. As a first approach a fully connected feed-forward Multi-Layer Perceptron (MLP) model with one hidden layer is used. It is intended that by choosing such a standard feed-forward MLP as approximation model no previous knowledge on the considered topology problem is included, yet assuming a sufficient number of hidden neurons it may model continuous and smooth non-linear functions. Furthermore, the implementation and parameter optimization are straight forward, and the relative simplicity of the model preserves a traceable complexity in the resulting TOPS approach.

Mathematically, a MLP with a single hidden layer is described as:

$$S_{\boldsymbol{\theta}}(\mathbf{s}_i) = g_o(\boldsymbol{\theta}_2^T \cdot g_h(\boldsymbol{\theta}_1^T \cdot \mathbf{s}_i)) \ , \tag{4.9}$$

with the weights $\boldsymbol{\theta}_1$ connecting input neurons with hidden neurons and the weights $\boldsymbol{\theta}_2$ connecting the hidden neurons with the single output neuron, with $\boldsymbol{\theta}^T = [\boldsymbol{\theta}_1^T \ \boldsymbol{\theta}_2^T]$. The activation function is designated g_h for the hidden neurons and g_o for the output neuron, respectively. Each neuron also has an additional bias as input. For H hidden neurons and J LSFs, the number of weights is $\Theta = (J+2) \cdot H + 1$. It is important to distinguish between the topology of the neural network model and the topology of the design. Since the neurons and connections within the network are fixed, the topology of the network does not change. Throughout this work, the term "topology" only refers to the topology of the design (i.e., the mechanical structure), which is to be optimized. An illustration of the MLP processing the LSF for an element is depicted in Fig. 4.8. It shows how, for each element in the design space, the LSFs are used as input to the network model. The model is processing the features and provides the update signal for the element.

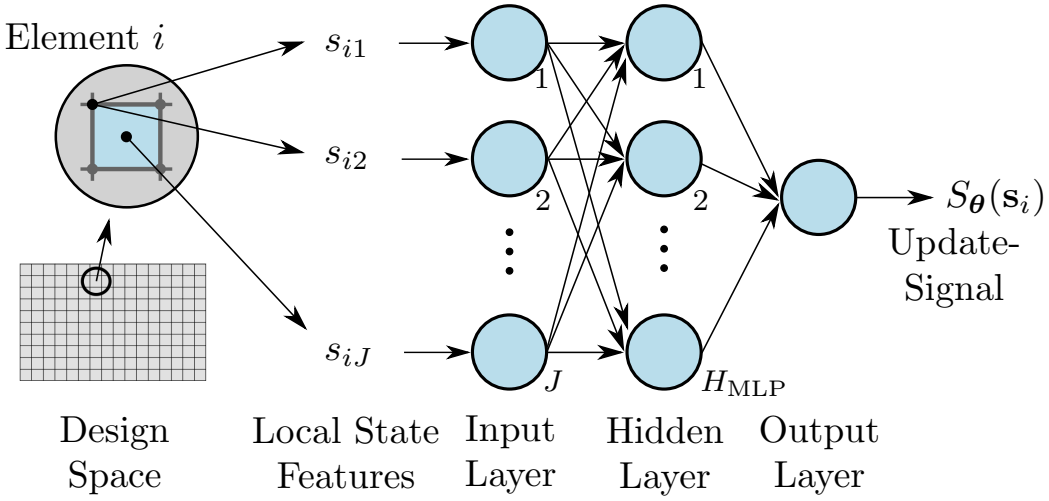


Figure 4.8: A feed-forward Multi-Layer-Perceptron with one hidden layer as update-signal model, processing Local State Features as input. A single model provides update-signals for all elements in the design space.

The generic topology optimization approach with neuro-evolution is referred to as NE-TOPS[3]. Results using NE-TOPS on a reference problem are presented in Sec. 5.2.

The proposed MLP model is a smooth function in the space of LSFs. However, it is possible that a high number of hidden units is required for adequate modelling of a non-linear function. The potentially high number of variables increases the computational cost of the evolutionary search. Therefore, rather than using a universal regression model, a different model, motivated from the perspective of a lower dimensional representation, is proposed in the next section.

4.3.2 Piecewise-Constant Model

In this section, another modelling approach is introduced that is motivated by the perspective of a lower dimensional representation of the structure, based on its state. It exploits the assumption that when elements with similar state are changed, they also have a similar effect on the objective function.

If (4.2) holds, then elements that have the same local state, also have

[3]In this thesis, several TOPS variants are proposed. For consistency of the naming convention, the variant is indicated by a prefix. This terminology deviates slightly from that used in publications. In [14, 15] the approach is referred to as Neuro-Evolutionary Topology Optimization (NETO).

the same sensitivity value. Accordingly, all elements having the same LSF vector have the same value of the update-signal, since $S_{\boldsymbol{\theta}}(\mathbf{s}_i) = S_{\boldsymbol{\theta}}(\mathbf{s}_j)$ if $\mathbf{s}_i = \mathbf{s}_j$. For the modelling approach in this section we additionally assume that several elements with *similar* LSFs can be assigned the *identical* sensitivity, i.e., when they are neighbours in LSF space. Formally, the model assumes $S_i(\mathbf{s}_i) = S_j(\mathbf{s}_j)$ if some distance measure $\text{dist}(\mathbf{s}_i, \mathbf{s}_j)$ is small.

Each element i relates to a point in LSF space by its LSF vector $\mathbf{s}_i \in \mathbb{R}^J$, with the number of LSFs J. This results in an unlabelled data set of LSF space samples $\mathcal{V} = \{\mathbf{s}_1, \ \ldots, \ \mathbf{s}_N\}$ which is illustrated in Fig. 4.9**(a)** and **(b)** for a two-dimensional LSF space.

This enables a *state based representation* by grouping elements that are assumed to have similar influence on the objective function, for instance elements that are close to each other in LSF space. By choosing a user-defined number of characteristic LSF vectors, a lower dimensional representation based on the structural state is defined, without geometric assumptions. Then, elements that have similar LSF can be represented by individual optimization variables.

Concretely, a simple form of vector quantization is applied: The data set $\mathcal{V}$ is mapped to a finite set of indices, based on P prototype elements. The LSF vectors $\mathbf{c}_l \in \mathcal{V}$, $l \in \{1, \ldots, P\}$ of the prototypes define clusters in LSF space. A mapping Ψ can be defined that maps each sample in data set $\mathcal{V}$ to an index, as

$$\Psi_{\mathcal{V} \to \{1,\ldots,P\}} : \mathbf{s}_i \to \zeta_i = \arg \min_{l \in \{1,\ldots,P\}} (\text{dist}(\mathbf{s}_i, \mathbf{c}_l)) \ ,$$

with the Euclidean distance metric $\text{dist}(\mathbf{s}_i, \mathbf{c}_l)$. The expression $\arg\min$ returns the value of l that minimizes the distance. Thus, each sample in $\mathcal{V}$ is assigned to a cluster with index $\zeta_i = \Psi(\mathbf{s}_i)$.

This relates to a subdivision of the LSF space in Voronoi cells. In contrast to existing work [155] the Voronoi cells are defined in LSF space and not within the Cartesian coordinates of the design space. An example for the resulting clusters in LSF space and the corresponding elements in the design space is shown in Fig. 4.9**(b)** and **(c)**, respectively. With random prototypes, each point in LSF space is assigned to one cluster of LSF points, highlighted by different colours in the figure. The elements represented by the clusters of LSF vectors are be mapped back to the design space, where they are not necessarily connected.

The number of prototypes can be chosen by the user, depending on the desired precision of the quantization. In the limit case $P \to N$, in which the number of prototypes approaches the total number of elements, each

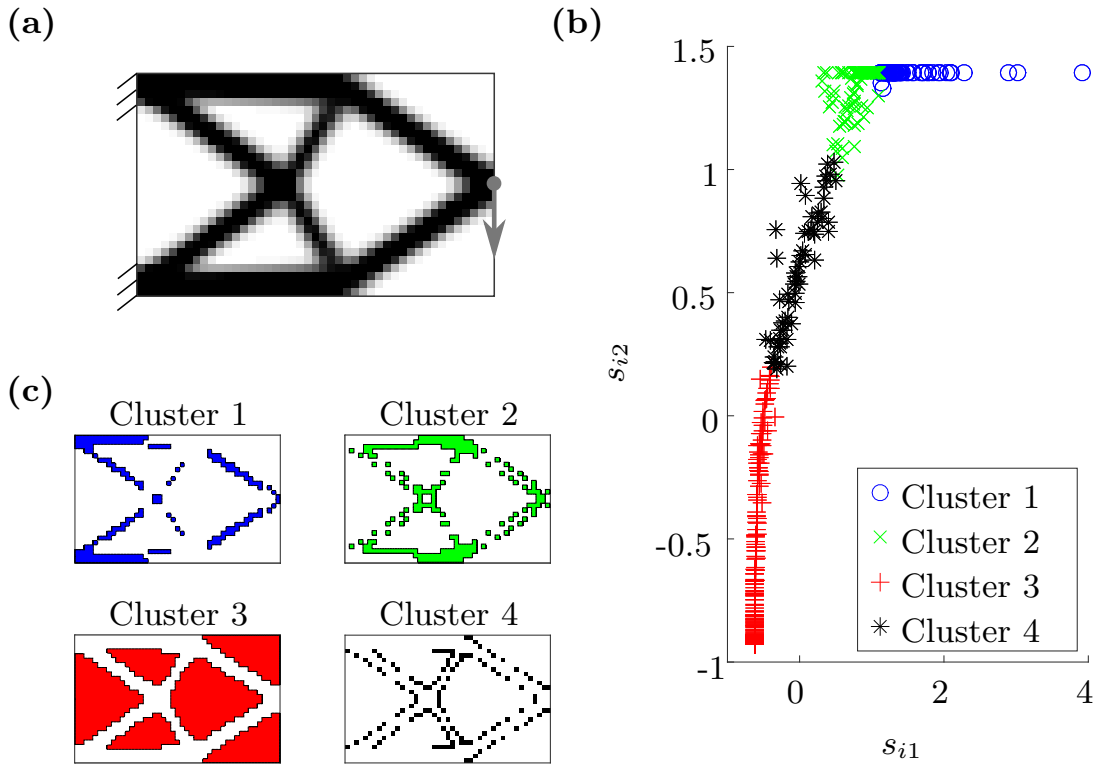


Figure 4.9: Example for: **(a)** Density distribution of cantilever structure subject to static load, **(b)** corresponding points in a two-dimensional LSF space, where s_{i1}, s_{i2} are normalized elemental strain energy and density, respectively, with clusters by choosing 4 prototype elements, and **(c)** resulting groups of elements in the design space.

element becomes a prototype. An advantage is that the number of clusters is decoupled from the number of elements in the finite element mesh, thus preferably $P << N$. This is important since the search space dimensionality is critical in terms of performance and cost of an evolutionary search.

The prototypes represent the structure depending on its state. By assigning an individual optimization variable to each prototype, a vector $\boldsymbol{\theta} = [\theta_1 \dots \theta_P]^T \in \mathbb{R}^P$ of search variables for the evolutionary optimization is obtained. The resulting update-signal can be stated as:

$$S_{\boldsymbol{\theta}}(\mathbf{s}_i) = \theta_{\Psi(\mathbf{s_i})} \tag{4.10}$$

Since the proposed representation of the structure is based on the state, which is a function of the material distribution, i.e. $\mathbf{u} = \mathbf{u}(\boldsymbol{\rho})$, any change in the structure will result in a change of the state. The new state leads to

different cluster assignments of the elements and eventually it is necessary to determine new prototypes that account for the change and reduce the quantization error. Therefore, it makes sense to only change the structure gradually, as it is done in the TOPS by the OC-update.

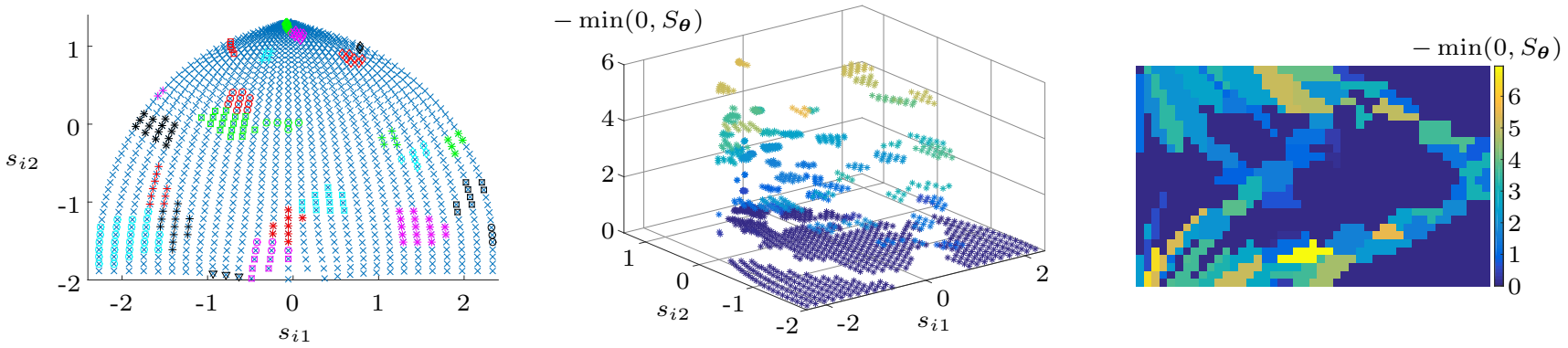
Specifically, an update-signal is assigned to each element prototype and its cluster of elements, resulting in a piecewise-constant function in LSF space. When this function is mapped back to the design space, it can be used to determine the increase or decrease of the density in the corresponding elements. An example illustrating the idea is shown in Fig. 4.10.

Since the update-signal model is composed of local zero-order models we designate the approach Topology Optimization by Predicting Sensitivities with Piecewise Constant Model (PCM-TOPS). Results on a reference problem are presented in Sec. 5.2.

When using PCM-TOPS, the first prototypes are determined and the elements are assigned to the respective cluster, in the initial learning step of the TOPS flow in Fig. 4.5. For every new training event, new prototypes are determined based on the changed structural state. By defining new prototypes, the representation is adapted to the changed material distribution within the design space. The adaptation decreases the quantization error, induced by the clustering. This facilitates a low number of prototypes and a feasible dimensionality for the evolutionary search.

Each time a retraining of the model is initiated, the prototypes are randomly chosen. Ideally, elements with the same sensitivity should be clustered. Accordingly, prototypes should be chosen so that elements with similar sensitivity are grouped together. However, the actual sensitivity is unknown and hence the ideal clusters as well. A random choice has the advantage that areas in LSF space with high density on average contribute more centre points, such that the groups of elements are of roughly equal size. Also, a random choice increases the variability of the representation, any good set of prototypes will be reused, while an unsuitable choice will be replaced quickly. Furthermore, the random choice is algorithmically consistent with the supervised approach introduced in the next section Sec. 4.4.

The PCM approach helps to understand the interesting transition from a state-based representation of the structure to a sensitivity prediction. In this case, modelling the sensitivities corresponds directly to a representation of groups of elements. As long as an active volume constraint is considered, any optimization of the densities will necessarily optimize the change of the elements, since for each elemental density that is reduced at



(a) **(b)** **(c)**

Figure 4.10: Example illustration of LSF space and update-signals: **(a)** Points in LSF space spanned by two components of the LSF vector, highlighting 25 of 150 clusters; **(b)** Presentation of the optimized piecewise-constant update-signal added as a third dimension orthogonally to the LSF space from **(a)**; **(c)** Presentation of the update-signal from **(b)** mapped back to the design space. For the presented plots, values of the update-signal larger than zero are cut and the result is inverted.

least one other needs to be increased. In consequence, an improvement of the structure can be typically expected when the optimization yields an accurate sensitivity model.

4.3.3 Optimization with CMA-ES

A standard, state-of-the-art evolutionary optimizer is applied to tune the parameters of the update-signal model as in (4.9) or (4.10). The weights of the neural network model in Sec. 4.3.1 are continuous as well as the parameters of the piecewise-constant model in Sec. 4.3.2. Therefore, it makes sense to apply an evolutionary strategy for optimization.

In the considered scenario of evolving weights of a neural network, the evolutionary optimization has similarities to a reinforcement learning approach of a controller. For the reinforcement tasks related to explicit policy learning methods, CMA-ES has shown to be a well-performing and robust optimizer. In direct policy learning, parameters of a function are searched explicitly. The goal is to maximize the reward and agent obtains for its actions that it takes depending on its state. Few comparisons can be found in the literature, however, it was demonstrated on the pole-balancing task that the Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) [76, 78] achieves very good results and is additionally more robust when compared to policy gradient methods [81].

As introduced in Sec. 2.3.1, CMA-ES is one of the best currently existing optimization methods for real-valued problems. It is a version of evolution strategies that utilizes the information collected by mutations in the previous iterations to estimate a covariance matrix for the mutation operator. This approach is especially useful for industrial optimization problems since it is able to find improved solutions with a relatively low number of evaluations [21]. Therefore, CMA-ES is also expected to perform well in case of the neural network and the PCM.

In fact, when applied to the optimization of directly encoded artificial neural network models, the CMA-ES has shown to be a competitive approach for the task of neuro-evolution with fixed network topologies [70, 82, 93, 106]. CMA-ES optimization of fixed network topology controllers outperforms several topology evolving methods and developmental neuro-evolutionary approaches often perform even below random walk approaches [70, 82, 93]. Fine-tuning the approaches according to specific results may conceivably produce more sophisticated topology-evolving and developmental approaches. For general tasks, it is more efficient to use a general purpose optimizer, such as CMA-ES, which is very efficient on a

large number of multi-modal test functions [77]. Below, we apply CMA-ES for addressing problem (4.8).

4.3.4 Computational Flow

For NE/PCM-TOPS, two optimization loops can be distinguished. The outer loop is the actual topology optimization as described in Sec. 4.2. For tuning of the update-signal model, an inner loop with the CMA-ES optimization of the update-signal is started. The objective (respectively fitness) of a model is defined as the objective (4.8) of the structure after the material distribution update. Fitness, then, is computed by emulating a single topology optimization update step, consisting of filtering and OC-update as described in Sec. 4.2.

The computational flow of the described NE/PCM-TOPS algorithm is depicted in Fig. 4.11. It extends Fig. 4.6(b) by specifying the internal loop for the model learning step. In this step, the evolutionary optimization

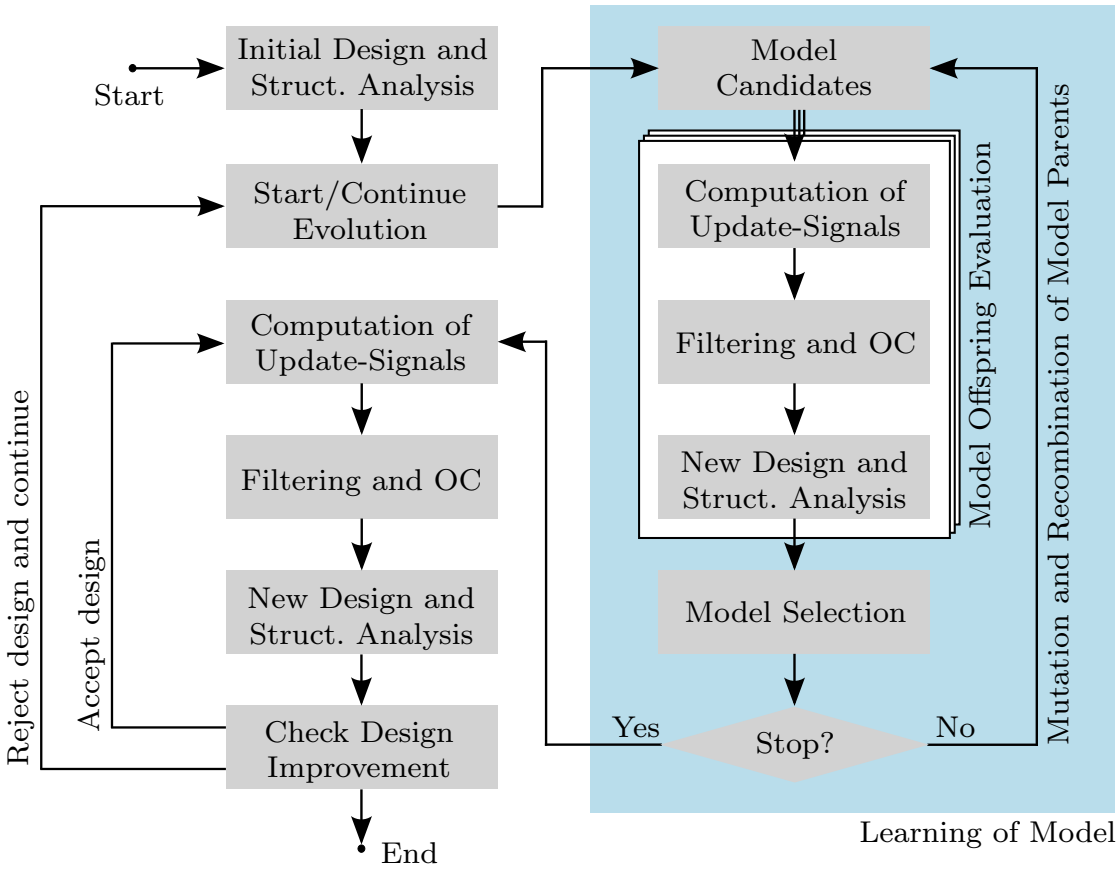


Figure 4.11: The computational flow for TOPS with explicit evolutionary tuning of the update-signal model with CMA-ES.

takes place.

During the evolution, as expressed by (4.8), the more the objective function of the new design is improved, the better the update-signals substitute the sensitivities and, thus, the better the model parameters are. The best models are selected as parents and a new set of model candidates is provided by the variational operators of the CMA-ES optimizer. This inner loop is run until a convergence criterion, or a maximum number of function evaluations, is reached. The optimized update-signal model is used in the outer topology optimization loop to perform a persistent design update, potentially for several iterations. When the learning step is reached again, the evolution is conducted based on the changed set of LSFs. As described in Sec. 4.2.1, an improvement threshold restricts the activation or continuation of evolutionary optimization runs. Only when the previously trained model is incapable to improve the structure, is the CMA-ES run again.

4.4 Sampling and Supervised Learning

As introduced in Chap. 2.2, the density-based topology optimization methods are classically utilizing analytical sensitivity information to perform a gradient descent to a local optimum. The previous section investigates the approach of replacing these sensitivities by a heuristic update-signal obtained from explicit evolutionary parameter optimization. An alternative approach is to sample the sensitivity by finite differences.

Under a finite differentiation approach, an estimate of the gradient is obtained by disturbing variables by a finite value under the assumption of local linearity. A classical naive, finite differentiation approach will however, at least require additional N evaluations for each design point. For small problems, such a naive, finite differentiation can be more efficient than an optimization with evolutionary algorithms. This case has been pointedly observed by Sigmund [170] with respect to the work of Wu and Tseng [211]. Albeit they considered a small mesh, the proposed modified, binary differential evolution with grid-based representation required more finite element evaluations than a finite differencing approach. For an EC approach with grid-based representation, this is problematic, since its application is limited to small mesh sizes anyway. In industrial topology optimization applications, however, the number of design variables can reach tens of thousands up to millions. The computational cost associated with the high number of finite element analyses renders naive, finite differentiation approaches infeasible.

It should be noted, again, that for NE-TOPS and PCM-TOPS approaches, as presented in the previous section, the number of variables in the evolutionary optimization is decoupled from the mesh resolution. Yet, treating the learning of the update-signal as black-box parameter optimization will tentatively result in high computational cost. Also, there is no guarantee that the update-signals will be related to the sensitivity, as any signals that improve the structure will be selected by the CMA-ES. Still, the more the model represents the actual sensitivity, the more the results will resemble those of a gradient-based optimization, providing a clearly understandable theoretical explanation for the structural improvement. Hence, when assuming that sensitivity information can be obtained from finite differentiation, this opens a way to replace evolutionary optimization by a more efficient machine learning approach.

Concretely, it is possible to train a regression model based on sampling the sensitivity of a subset of the variables. This data is obtained from finite differentiation, combined with the LSFs associated with each design variable. Thus, the problem of generic topology optimization can be considered as a supervised regression learning task, which is a standard problem in the field of machine learning. In contrast to unsupervised learning, supervised learning requires a set of training examples with known output values. The sensitivity-sampling approach will be expounded in the next section.

4.4.1 Sensitivity Estimation by Finite-Difference Sampling

A finite difference (FD) is a small perturbation added to a design variable in order to observe the change of the objective function (e.g. [55, 65, 175]). A one sided forward finite difference is defined as:

$$\frac{\Delta F_i}{||\boldsymbol{\Delta\rho}_i||} = \frac{F(\boldsymbol{\rho}+\boldsymbol{\Delta\rho}_i)-F(\boldsymbol{\rho})}{\epsilon} \approx \frac{\partial F}{\partial \rho_i} \ , \tag{4.11}$$

where $\boldsymbol{\Delta\rho}_i = [\Delta\rho_{i1} \ \dots \ \Delta\rho_{ij} \dots \ \Delta\rho_{iN}]^T$ is the perturbation vector, consisting of zeros, except for the i-th element, where it contains the finite difference step length ϵ.

$$\Delta\rho_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ \epsilon & \text{if } i = j \end{cases} \ .$$

The accuracy of the estimate can be increased by using a two sided perturbation, i.e. central finite differentiation:

$$\frac{\Delta F_i}{||\boldsymbol{\Delta\rho}_i||} = \frac{F(\boldsymbol{\rho} + \boldsymbol{\Delta\rho}_i) - F(\boldsymbol{\rho} - \boldsymbol{\Delta\rho}_i)}{2\epsilon} \approx \frac{\partial F}{\partial \rho_i} \quad . \tag{4.12}$$

Accordingly, N function evaluations are required for a forward finite differentiation approach, and $2N$ function evaluations for a central finite differentiation approach, to obtain sensitivity information for all design variables. The problem of approximating the gradient of an N-dimensional function can be formulated in general as a linear regression problem, i.e. fitting a hyper plane to the function, requiring $N + 1$ sample points.

The "Learning of Model" step in the general TOPS approach as in Fig. 4.5 can be implemented by the use of finite differentiation. Conventionally, the required number of finite element analyses per iteration, when using gradient-estimation via finite differentiation is at least $N + 1$. Yet, when applying the idea of the update-signal model from Sec. 4.1.3, it is *not* necessary to perform finite differentiation for all N variables to obtain update-signals for all variables.

An update-signal model that reflects the actual sensitivities of the design variables, can be obtained by sampling a subset of the elemental sensitivities. A training sample is obtained by approximating the sensitivity by finite differentiation in (4.11) or (4.12) for a single design variable i. This yields a sensitivity sample $\left(\mathbf{s}_i, \frac{\Delta F}{\Delta \rho}\right)$. If this is repeated for T design variables, a set of samples consisting of the LSFs of an element and its estimated sensitivity is collected:

$$\{(\mathbf{s}_1, y_1)\,,\ (\mathbf{s}_2, y_2)\,,\ \ldots,\ (\mathbf{s}_T, y_T)\} \quad ,$$
$$\text{with } y_i = \frac{\Delta F_i}{\Delta \rho} \quad .$$

The set of training samples is used to train a regression model with a supervised approach. This update-signal model can predict the sensitivities of other design variables based on the LSF. Figure 4.12 illustrates this idea.

Since the computational cost for training a model and predicting sensitivities for the remaining design variables is assumed to be much smaller than that for performing finite element analysis simulations for all of the remaining variables, this is a more efficient approach than estimating the gradient by pure finite differentiation.

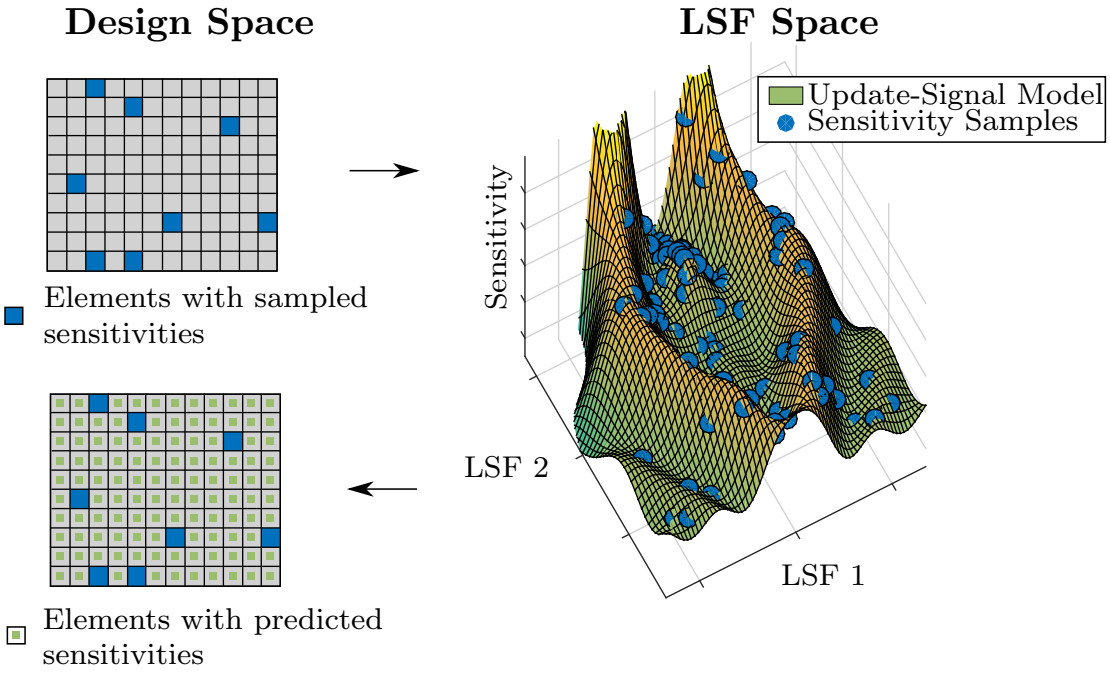


Figure 4.12: The idea of TOPS with sensitivity-sampling: For a subset of the elements in the design space, the sensitivity is found using a FD technique. Based on these samples, the update signal model is trained and used to predict sensitivities for the remaining majority of elements in the design space.

Compared to the tuning of the model with evolutionary optimization, the advantage is a reduction of the computational cost for training the model, since the sensitivity target function is modelled explicitly. Once again, it is important to emphasize that the usage of features associated with the design variables differentiates the approach from typical surrogate-assisted optimization (e.g. [65]), estimation of distribution algorithms, such as CMA-ES, or generally derivative-free methods. For TOPS, not a model for the objective function is learned but a model of the local sensitivities of the variables, based on the LSFs. This provides an alternative to computing finite differences for all design variables.

Since the design is changing continuously during the optimization, it is possible that samples obtained from earlier iterations do not contribute to the prediction, or even contradict information gathered from the more current designs. Thus, it is advisable to use only samples obtained from a number of previously sampled designs.

4.4.2 Aggregated Sensitivity-Sampling

In the previous section, it was proposed to determine the update-signal model by learning from the explicit sampling of the design variables' sensitivities. Although more efficient than tuning of the model parameters by evolutionary optimization, the approach relies on the assumption that the sensitivity of a variable can be estimated sufficiently precise by finite differentiation techniques. Yet, it is possible that this assumption is not true for industrial optimization problems, especially those involving black-box simulations. An example is the case of crashworthiness topology optimization, which is the application considered in Chap. 6. The reason is that crashworthiness simulations involve significant amounts of numerical noise.

Theoretically, the sensitivity is estimated correctly for $\epsilon \to 0$. Practically, it is limited by the accuracy of the considered function, i.e. the underlying simulation. The accuracy of the simulation result is limited at least by the mesh resolution that is still feasible in terms of required computational time. Hence, the effect of modifying a single element is hard to measure accurately, and any attempt usually contains numerical noise in the sensitivity estimation. This problem is alleviated when the variable is varied with a larger finite difference step, up to the complete variable bounds; however, in this case, the linearity assumption might be violated, causing again an imprecise estimation of the sensitivity.

In order to alleviate the problem of numerical noise, the TOPS algorithm can be extended by further exploiting the assumption on the relation between the LSFs and the sensitivity, as it was done for the PCM in Sec. 4.3.2. Again, we additionally assume that elements with *similar* LSFs also have a similar sensitivity, where similarity is measured by a distance measure in LSF space.

Then, it is possible to perform a concurrent finite difference step, i.e. a simultaneous perturbation [175] for several variables, which have similar LSFs. The aggregated change in the objective function can then be mapped to an average sensitivity of the involved elements. Concretely, instead of estimating the sensitivity of ρ_i by a finite difference step as in (4.11), the sensitivity can be estimated by an *aggregated* sampling step. For this purpose we introduce the aggregated finite difference vector

$$(\boldsymbol{\Delta\rho}_i)^{\text{aggr}} = [\Delta\rho_{1j}\ \Delta\rho_{2j}\ \dots\ \Delta\rho_{Nj}]^T,\ \Delta\rho_{ij} = \begin{cases} \epsilon & j \in \mathcal{G}_i \\ 0 & j \notin \mathcal{G}_i \end{cases},$$

where $\mathcal{G}_i$ is a set that contains the indices for the $N_{\mathcal{G}}$ elements, which are

closest to element i in LSF space. With the Euclidean distance in LSF space we can calculate the length $||(\boldsymbol{\Delta\rho}_i)^{\text{aggr}}|| = \sqrt{N_\mathcal{G}}\epsilon$.

Based on this finite difference, the approximation of the partial derivative can be found as:

$$\frac{(\Delta F_i)^{\text{aggr}}}{||(\boldsymbol{\Delta\rho}_i)^{\text{aggr}}||} = \frac{F(\boldsymbol{\rho} + (\boldsymbol{\Delta\rho}_i)^{\text{aggr}}) - F(\boldsymbol{\rho})}{\sqrt{N_\mathcal{G}}\epsilon} \approx \frac{\partial F}{\partial \rho_i} \quad .$$

In this way, a sensitivity sample $(\bar{\mathbf{s}}_i,\ (\Delta F_i)^{\text{aggr}} / \,||(\boldsymbol{\Delta\rho}_i)^{\text{aggr}}||)$ is obtained, where $\bar{\mathbf{s}}_i$ is the average of all LSF vectors belonging to the elements in $\mathcal{G}_i$. If we assume that the local linearity assumption for F is true, then $(\Delta F_i)^{\text{aggr}}$ increases with $N_\mathcal{G}$, i.e. the changes of the objective function for perturbation along one axis ΔF_i are summed up. However, an additional error is introduced by the distance of the LSF samples of $\mathcal{G}_i$ in LSF space.

The new parameter $N_\mathcal{G}$ needs to be specified by the user. Case $N_\mathcal{G} = 1$ is the original supervised approach from section 4.4.1. In case $N_\mathcal{G} > 1$, aggregated sensitivities are used. This can be designated as Topology Optimization by Predicting Aggregated Sensitivities (TOPAS). For larger values of $N_\mathcal{G}$ more elements will be aggregated and the sensitivity estimate will become less precise, since the elements will have larger and larger distances in feature space. Yet, for larger $N_\mathcal{G}$, also the effect of the noise of the objective function will decrease, relatively to a FD step as in (4.11).

Using aggregated sampling, the structure is changed more than for simple forward differentiation. The overall response of the structure is more clear compared to changing a single element, but can still be related to certain values of the LSF vector. An intuitive example is to imagine that all elements that have a similar strain energy density are slightly increased in the material density. Intuitively speaking, instead of doing a finite difference step for an element with a certain energy, an aggregated finite difference step is done for all elements with the same amount of energy and each element is assumed to contribute equally to the change in the objective function. This is very similar to the idea of clusters in LSF space as illustrated in Fig. 4.9, with the difference that the group for the aggregated sampling are simply the $N_\mathcal{G}$ closest element neighbours to element i.

Compared to increasing the finite difference step by factor $N_\mathcal{G}$, the aggregated finite differentiation reduces the Euclidean distance to the current design point by factor $\sqrt{N_\mathcal{G}}$. Thus, the assumption is made that linearity is true close to the design point in the sense of Euclidean distance (i.e. changing in several dimensions in parallel), rather than for a larger change along one dimension.

As argued, the aggregated sampling scheme can be beneficial for noisy objective functions. If we assume that the objective function is subject to additive noise:

$$\tilde{F} = F + z \ ,$$

with a random number $z \sim \mathcal{N}(0,\, \sigma_F^2)$ from a normal distribution with standard deviation σ_F. It holds:

$$\frac{\Delta\tilde{F}_i}{\epsilon} = \frac{\tilde{F}(\boldsymbol{\rho} + \boldsymbol{\Delta\rho}_i) - \tilde{F}(\boldsymbol{\rho})}{\epsilon} = \frac{F(\boldsymbol{\rho} + \boldsymbol{\Delta\rho}_i) - F(\boldsymbol{\rho})}{\epsilon} + \frac{z'}{\epsilon} \ ,$$

with $z' \sim \mathcal{N}(0,\, 2\sigma_F^2)$. Using the aggregated sampling, we obtain

$$\begin{aligned}\frac{(\Delta\tilde{F}_i)^{\text{aggr}}}{||(\boldsymbol{\Delta\rho}_i)^{\text{aggr}}||} &= \frac{\tilde{F}(\boldsymbol{\rho} + (\boldsymbol{\Delta\rho}_i)^{\text{aggr}}) - \tilde{F}(\boldsymbol{\rho})}{\sqrt{N_{\mathcal{G}}}\epsilon} \\ &= \frac{F(\boldsymbol{\rho} + (\boldsymbol{\Delta\rho}_i)^{\text{aggr}}) - F(\boldsymbol{\rho})}{\sqrt{N_{\mathcal{G}}}\epsilon} + \frac{z'}{\sqrt{N_{\mathcal{G}}}\epsilon}\end{aligned}$$

such that the noise is reduced with increasing group size $N_{\mathcal{G}}$.

4.4.3 Sensitivity Regression Model

A regression model maps a vector of input variables/features to a continuous output variable. Regression models are a class of statistical learning models, typically obtained by data-driven supervised machine learning methods. Theoretical background can be found in the literature, e.g. [32]. In this thesis, the supervised TOPS approach is applied in conjunction with linear regression and Support Vector Regression (SVR) [173, 194].

Linear regression models are used as a basic model in which the learning process and the result is rather easily traceable. It identifies linear relations between the features and the target. The linear regression model is defined as:

$$S_{\boldsymbol{\theta}}(\mathbf{s}) = \theta_0 + \theta_1 s_1 + ... + \theta_J s_J = \boldsymbol{\theta}^T \begin{bmatrix} 1 \\ \mathbf{s} \end{bmatrix} \ , \tag{4.13}$$

with the model parameters $\boldsymbol{\theta}$ and the J-dimensional feature vector $\mathbf{s}$. The generic topology optimization approach with linear regression model is referred to as LIN-TOPS.

The support vector learning method is a standard tool of machine learning as it has shown excellent performance in many regression and classification tasks and is easy to use. In contrast to linear regression, SVR

is suitable to model non-linearities (for instance when no expert LSF are available). The SVR model [194] can be stated as:

$$S_{\boldsymbol{\theta}}(\mathbf{s}) = \sum_{v=1}^{V} \theta_l \hat{k}\left(\mathbf{s}_l,\mathbf{s}\right) + \theta_0 \ ,$$

where $\mathbf{s}_l$ are the V support vectors and $\hat{k}\left(\mathbf{s}_l,\mathbf{s}\right)$ is the kernel function, representing a dot product of a transformation of the input vector in a higher dimensional feature space. The generic topology optimization approach with SVR is referred to as SVR-TOPS.

The detailed description of the models is provided in App. A.2. In the context of this thesis, the number of training samples is relatively low so that over-fitting can be a problem and regularization methods are of importance and also briefly described.

4.4.4 Computational Flow

Based on the idea described in the preceding sections, the computational flow for the TOPS algorithm using supervised learning is proposed. Figure 4.13 shows the computational flow of TOPS with sensitivity-sampling, in which the "Learning of Model" step is elaborated.

Initially, we assume no training samples are available. After the initial analysis, the following steps are iterated:

1. A random indices subset of the design variables of size $M < N$, for which the sensitivity will be sampled, is chosen. No design variable is selected more than once for the current design.

2. The sensitivity with respect to the objective function and the current design is estimated separately for the set of chosen design variables by performing finite difference steps.

3. The sampled sensitivities and the corresponding features are stored.

4. The parameters of a new update-signal model are trained with a learning algorithm, based on the stored samples.

5. The update-signal model predicts sensitivities for all design variables which were not sampled by finite differentiation in step 1.

6. The predicted sensitivities are filtered and the OC-update is applied.

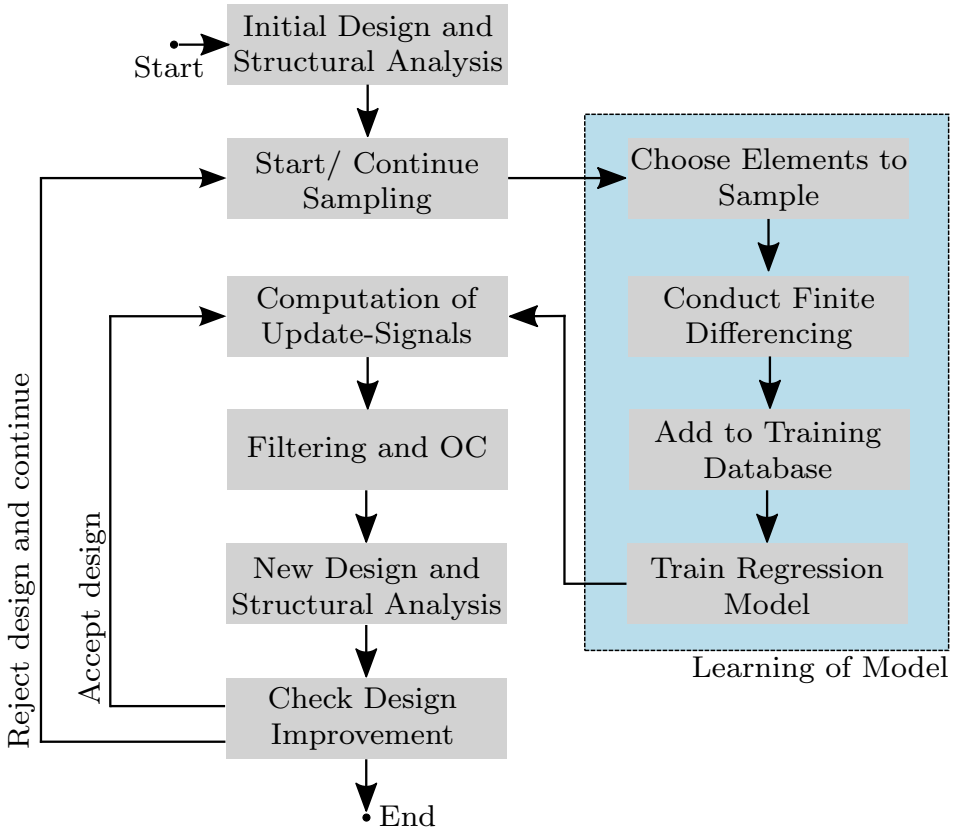


Figure 4.13: The computational flow for TOPS based on supervised learning of regression models with finite difference sampling of sensitivities.

7. The quality of the new design candidate is assessed by a finite element analysis.

8. The improvement threshold as described in Sec. 4.2.1 is checked. If the design improvement is larger than the improvement threshold, the design candidate becomes the current design and the algorithm continues in step 5. Otherwise the design candidate is rejected and either the optimization is stopped, or additional samples are taken for the current design, i.e. the algorithm continues in step 1.

4.5 Summary of Contribution

This chapter presents a novel, generic topology optimization approach: Topology Optimization by Predicting Sensitivities (TOPS). TOPS includes a learning component, which enables the algorithm to adapt to the considered topology optimization objective function. This learning com-

ponent generates a model that determines the redistribution of material throughout the design space. Unlike conventional optimization approaches that are based on surrogate modelling, our unique approach introduces a systematic exploitation of simulation data: Local State Features (LSF) are the independent model inputs. The LSFs are obtained from structural analysis. They are specific with respect to each finite element as well as to the corresponding variables. As a result, the TOPS approach is more widely applicable than regular gradient-based methods, as it does not require pre-defined sensitivity formulations. It can also address problems for which no or very limited physical/mathematical understanding is available, for instance, black-box simulations.

The first model-learning method proposed for TOPS is explicit evolutionary learning. Since the number of model parameters is decoupled from the number of finite elements, a high dimensional mesh representation can be maintained. This is an advantage compared to typical approaches from the field of evolutionary computation, which require a coarse grid or a simplified representation. Furthermore, the proposed piecewise-constant model reveals an interesting connection between sensitivity prediction and a state-based representation. The second learning approach carved out in this chapter is supervised learning. Using supervised learning techniques, the model can be learned from finite differences. Potentially, it enables a significant decrease of computational cost, compared to a naive, finite differentiation approach, since only a subset of variables needs to be sampled.

The following chapter critically evaluates the generic topology optimization method with its various learning approaches. The goal is to show the conditions under which each of the approaches are able to reproduces a reference structure and to identify scenarios for application.

5 Studies on the Minimum Compliance Problem

This chapter evaluates the proposed generic topology optimization method and presents comprehensive results. Primarily, Topology Optimization by Predicting Sensitivities demonstrates its feasibility by reproduction of a known reference solution in empirical studies. The addressed reference for the experiments is compliance minimization, which is a standard topology optimization problem. From a research perspective, it serves as reference for testing novel methods. Secondly, the comparison of the different learning variants in various settings, yields valuable insights on important working principles and parameters. These findings will help to address more challenging applications, for instance in the subsequent chapter.

At first, the minimum compliance cantilever reference problem and the Local State Features of this linear elastic static problem are presented in Sec. 5.1. The remaining part of the chapter is composed of the experimental sections. Section 5.2 presents the results of update-signal learning based on explicit evolutionary optimization. It is followed by the results of the supervised update-signal learning in Sec. 5.3. The chapter is concluded with an overall discussion of the results in Sec. 5.4. Parts of this chapter are based on [9, 10, 14].

5.1 Reference Problem

As any new topology optimization algorithm, the TOPS approach requires validation on a reference problem. A standard topology optimization problem is linear elastic compliance minimization, for which well-known reference solutions and analytical sensitivity information are available (see Sec. 2.2.2). Hence, the learning target is known a-priori. An additional asset of this problem is moderate computational cost of the linear static state equation. It can be solved very efficiently when using existing solver implementations, a fact that enables statistical investigations at least for moderate mesh sizes.

The two-dimensional cantilever beam is one of the standard test problems agreed on in the topology optimization community. Accordingly, a solution for comparison is obtained easily and we chose it as reference for our studies. Design space and boundary conditions are shown in Sec. 2.2.2 in Fig. 5.1. The parameters for the test case are given in Tab. 5.1 unless otherwise specified. The load l is the non-zero component of the load vector l at the index of the centre node along the right edge of the design space. The symbols $r_{\min}$ and m have been introduced in Sec. 4.2, $\Delta F_{\min}$ in Sec. 4.2.1 and all others in Sec. 2.2, respectively

In the common formulation of the minimum compliance problem in (2.5), there are no additional constraints besides the volume constraint. Hence, for solving the problem with a standard gradient-based optimization, mainly a sensitivity formulation of the compliance c is required. The derivation of the adjoint sensitivities can be found in the literature (for instance [30]) or in App. A.1. It is possible to express the sensitivities of the compliance

$$c(\mathbf{u}(\boldsymbol{\rho})) = \mathbf{u}^T\mathbf{l}$$

based on the localized state information as:

$$\frac{\partial c(\boldsymbol{\rho})}{\partial \rho_i} = \frac{\partial c(\boldsymbol{\rho})}{\partial \rho_i}(\boldsymbol{\rho}_i, \mathbf{u}_i) = -p\rho_i^{p-1}\mathbf{u}_i^T\mathbf{K}_i\mathbf{u}_i \ , \tag{5.1}$$

where $\mathbf{u}_i$ is the elemental, nodal displacement vector, $\mathbf{K}_i$ is the elemental stiffness matrix, p is the SIMP penalization and ρ_i is the density. Since a regular mesh is considered, $\mathbf{K}_i = \mathbf{K}_0$ is constant and the sensitivities depend only on the local displacement vector and the density. The elemental stiffness matrix $\mathbf{K}_0$ is given in App. A.3. A homogenous distribution of material ($\rho_i = f, i = 1, \ldots, N$) is used as initial design guess. The resulting design when optimizing with analytical sensitivities (5.1), optimality

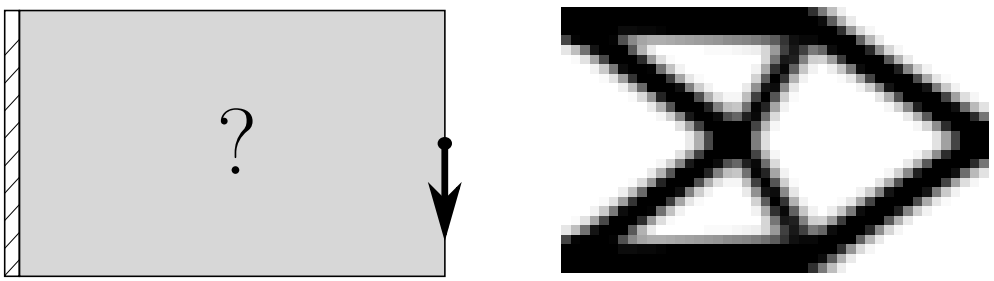


Figure 5.1: Minimum compliance cantilever reference problem: design space (left) and reference solution (right).

Table 5.1: Parameter settings for the plane stress minimum compliance cantilever reference problem.

Setting	Symbol	Value
Number of elements	N	$45 \times 28 = 1260$
Element size	-	1mm x 1mm
Load	l	1N
Target volume fraction	f	0.4
Young's modulus	E	1 N/mm^2
Poisson's ratio	ν	0.3
Penalization	p	3
Filter radius	$r_{\min}$	2.0mm
Move limit	m	0.2
Minimum improvement	$\Delta F_{\min}$	0.001

criteria update, and filtering of sensitivities, as proposed in Andreassen et. al. [5] is shown in Fig. 5.1.

For testing of the generic topology optimization, it is assumed that analytical sensitivities are unknown and a reference solution is obtained by gradient estimation using forward finite differentiation (as in (4.11)) for all design variables. The optimization stops after 21 iterations when $\Delta F \leq \Delta F_{\min} = 0.001$. The resulting solution corresponds almost exactly to the structure obtained by analytical sensitivities in Fig. 5.1. The obtained reference objective value is $c_{\text{ref}} = 54.45$Nmm and the required number of evaluations is $M_{\text{ref}} = 26{,}481$. These values and the corresponding structure form a baseline for comparisons in Sec. 5.2 and Sec. 5.3.

5.1.1 Linear Static LSF

Evaluation of the structure requires solving the linear static state equation (2.6) . Most experiments in this chapter consider two different LSF vectors that are obtained from this solution and represent different amounts of previous knowledge. The two LSF vectors are introduced in the following.

LSF Vector I $\mathbf{s}_i^{\text{I}} \in \mathbb{R}^{10}$ represents a test case, in which expert knowledge, such as e.g. intuition or (semi-)analytical information, provides at least one

strong feature for the problem. The components of $\mathbf{s}_i^{\mathrm{I}}$ are defined as:

$$s_{ij}^{\mathrm{I}} = \begin{cases} u_{ij} & \text{for } j = 1,2,\ldots,8, \\ \rho_i & \text{for } j = 9, \\ \mathrm{SED}_i & \text{for } j = 10 = J^{\mathrm{I}}, \end{cases} ,$$

with the elemental density ρ_i, nodal displacement components u_{ij} and the strain energy density SED_i as LSFs.

Biologically-inspired or intuition-based topology optimization algorithms that utilize the strain energy usually achieve acceptable results when applied to the optimization of minimum compliance structures. This is stressed by Sigmund et.al. in [171]: "*Therefore, simple schemes that add material where strain energy densities are high and subtract material where they are low will always work fine.*". The reason for this is that the strain energy density is proportional to the compliance sensitivity. In order to illustrate this, Figure 5.2 shows the sensitivity versus some LSFs obtained from analysing the initial structure of the reference test case. The linear relation of strain energy density and sensitivity is confirmed in the plot, while the relation is clearly non-linear for the nodal displacement LSFs.

Therefore, the elemental strain energy density can be considered as a very valuable LSF for the reference problem. It can be constructed from the basic features according to:

$$\mathrm{SED}_i = \rho_i^p \mathbf{u}_i^{\mathrm{T}} \mathbf{K}_0 \mathbf{u}_i / v_i \ ,$$

with unit elemental volume $v_i = 1$. This LSF represents human expertise about the problem, as it contains direct knowledge on element formulation and the material constants. It also contains implicit assumptions by selecting this feature over other physical information on the element, like e.g. strains or stresses.

However, in the test scenario, the TOPS approach does not have the a-priori information whether to remove or to add material when the strain energy density is high. Furthermore, the information on the elemental strain energy density is mixed with that of the other LSFs. A successful learning will nevertheless use it to improve the structure. Still, the learning task with LSF Vector I consists mainly of identifying the simple linear relation and serves as a basic validation of the approach.

For a more challenging learning task the strain energy density is omitted and only the basic LSFs as introduced in section 4.1.3 are used. The

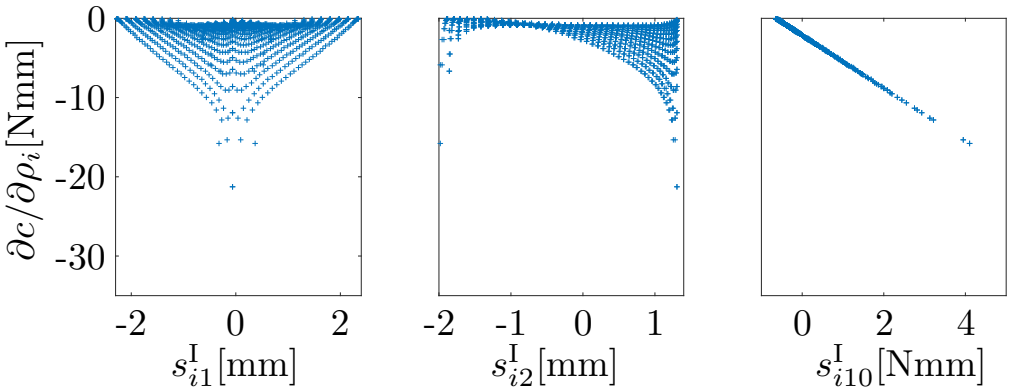


Figure 5.2: The relation of several LSFs of $\mathbf{s}^{\mathrm{I}}_i$ and corresponding elemental sensitivity shown for all elements of the initial structure. The sensitivity can take several values for a nodal displacement component $s^{\mathrm{I}}_{i1,2} = u_{i1,2}$, yet it is clearly related to the strain energy density $s^{\mathrm{I}}_{i10} = \mathrm{SED}_i$ by a linear function.

components of LSF Vector II $\mathbf{s}^{\mathrm{II}}_i \in \mathbb{R}^9$ are given as:

$$s^{\mathrm{II}}_{ij} = \begin{cases} u_{ij} & \text{for } j = 1,2,\ldots,8, \\ \rho_i & \text{for } j = J^{\mathrm{II}} = 9. \end{cases}$$

Without the strain energy density LSF, this choice of features does not include additional previous knowledge about the problem at hand and is generally available when a (displacement-based) finite element solver is applied for analysing the design. Assuming constant p and $\mathbf{K_0}$ the information on the energy is contained implicitly in $\mathbf{s}^{\mathrm{II}}_i$, however the learning task is significantly harder: In order to exactly obtain the information where to add and remove material, the non-linear, multi-variate polynomial in the basic LSFs from (5.1) has to be modelled. The difficulty can also be seen intuitively from the non-linear relation between LSF and sensitivity for the displacement features as shown in Fig. 5.2.

Before any learning is performed, all LSF vector components are normalized to zero mean and unit standard deviation.

5.2 NE/PCM-TOPS Experiments

In this section, several experiments on the minimum compliance reference cantilever problem are conducted to empirically evaluate the generic topology optimization approaches based on evolutionary optimization proposed in Sec. 4.3:

- Neuro-evolutionary Topology Optimization by Predicting Sensitivities (NE-TOPS),
- Piecewise-Constant Model Topology Optimization by Predicting Sensitivities (PCM-TOPS).

Both approaches are conducted using LSF Vector I and LSF Vector II with different model complexities. The number of prototypes P for PCM-TOPS is set to match the corresponding number of parameters of the network model for NE-TOPS, such that the CMA-ES is faced with the same search dimensionality Θ. The number of inputs to the network depends on the LSF Vector: Θ is different for LSF Vector I and II, since $J^{\mathrm{I}} > J^{\mathrm{II}}$. The parameter settings are specified in Tab. 5.2 and it is indicated if different values are chosen depending on the different LSF vectors. For each run, the CMA-ES population size is chosen based on the recommendations by Hansen and Ostermeier [78].

The initial model parameters are drawn from a uniform random distribution in $[-0.3,\ 0.3]$ for the neural network weights and biases $\boldsymbol{\theta}_{1,2}$ and in $[-1,\ 0]$ for the PCM prototype update-signals θ_{Ψ}. For some parameter configurations the model may be infeasible, because it renders a design update impossible with the OC-update (for example when outputs for all elements are identical). In such cases a very high objective value is assigned.

The methods are implemented in Matlab [189] and use the available code for the CMA-ES by Hansen[1]. For OC-update, filtering of sensitivities and FE solver, public code is used [5]. Each experiment was run for 30 different random seeds. As can be expected for algorithms with stochastic components, there is variance in the resulting solutions. Evaluation takes

[1] `https://www.lri.fr/~hansen/cmaes_inmatlab.html#matlab`

Table 5.2: Parameter settings for NE/PCM-TOPS experiments.

Setting	Symbol	Value
Number of hidden neurons	H	$1, 2, 5, 10, 15, 30$
Number of prototypes	$P^{(\mathrm{I})}(=\Theta^{(\mathrm{I})})$	$13, 25, 61, 121, 181, 361$
	$P^{(\mathrm{II})}(=\Theta^{(\mathrm{II})})$	$12, 23, 56, 111, 166, 331$
Evaluations per learning step	M	2000
Max. evaluations per iteration	$M^{\max}$	4000
Initial global step size	$\sigma^{(k=0)}$	0.3

this into account, e.g. by showing structures for best, mean and worst runs. Unless otherwise specified, if a structure or curve is shown or if it is indicated as "mean" this corresponds to the run that resulted in the solution with the compliance closest to the mean compliance of the solutions of the 30 trials. Quantitative results for all experiments are collected in App. B. For the conclusions drawn, statistical t-tests are performed based on the 5% significance level. In the following, the results for both LSF vectors are presented separately with a subsequent combined discussion.

5.2.1 Results LSF Vector I

In this section, results for LSF Vector I are described. An overview of the results in terms of compliance and evaluations is shown in Fig. 5.3. In terms of compliance, the two approaches show an opposing trend. For NE-TOPS the lowest mean compliance is obtained for lowest model complexities (one and two hidden neurons), while for PCM-TOPS the lowest mean compliance is obtained for the highest model complexity (361 prototypes). In terms of evaluations there is an increasing trend for both approaches. The results are further studied for the model complexities with lowest mean compliance.

For these model complexities, there is no statistical difference in terms of compliance between PCM- and NE-TOPS, yet in terms of evaluations NE-TOPS is much more economical (on average in total $1.85 \cdot 10^4$ opposed to $3.36 \cdot 10^4$ evaluations). Figure 5.4 shows the objective values versus the total number of function evaluations. NE-TOPS reaches a solution that outperforms the reference[2] after roughly 6,000 evaluations. The update-signal model is reused in some of the iterations. This shows up as vertical descents in the plot, since no new evaluations are required for learning. In this case, only singular evaluations are necessary to validate this improvement. This a clear advantage compared to learning a new update-signal

[2]Intuitively, it is expected that the best possible update-signals cannot outperform the reference obtained by finite differencing gradient, since naively it is expected that the update signal would take the same values in the best case. However, the evolutionary optimization can exploit random changes that modify the result towards a more discrete solution, thus reducing the amount of remaining intermediate (and penalized) densities compared to the reference. If the reference structure is post-processed to a zero-one structure, by applying a simple threshold to the densities that maintains the volume fraction, its compliance decreases from = 54.45Nmm to 47.13Nmm. When the same procedure is applied to the best NE-TOPS structure its compliance decreases from 51.18Nmm to 46.93Nmm, which is only marginally lower.

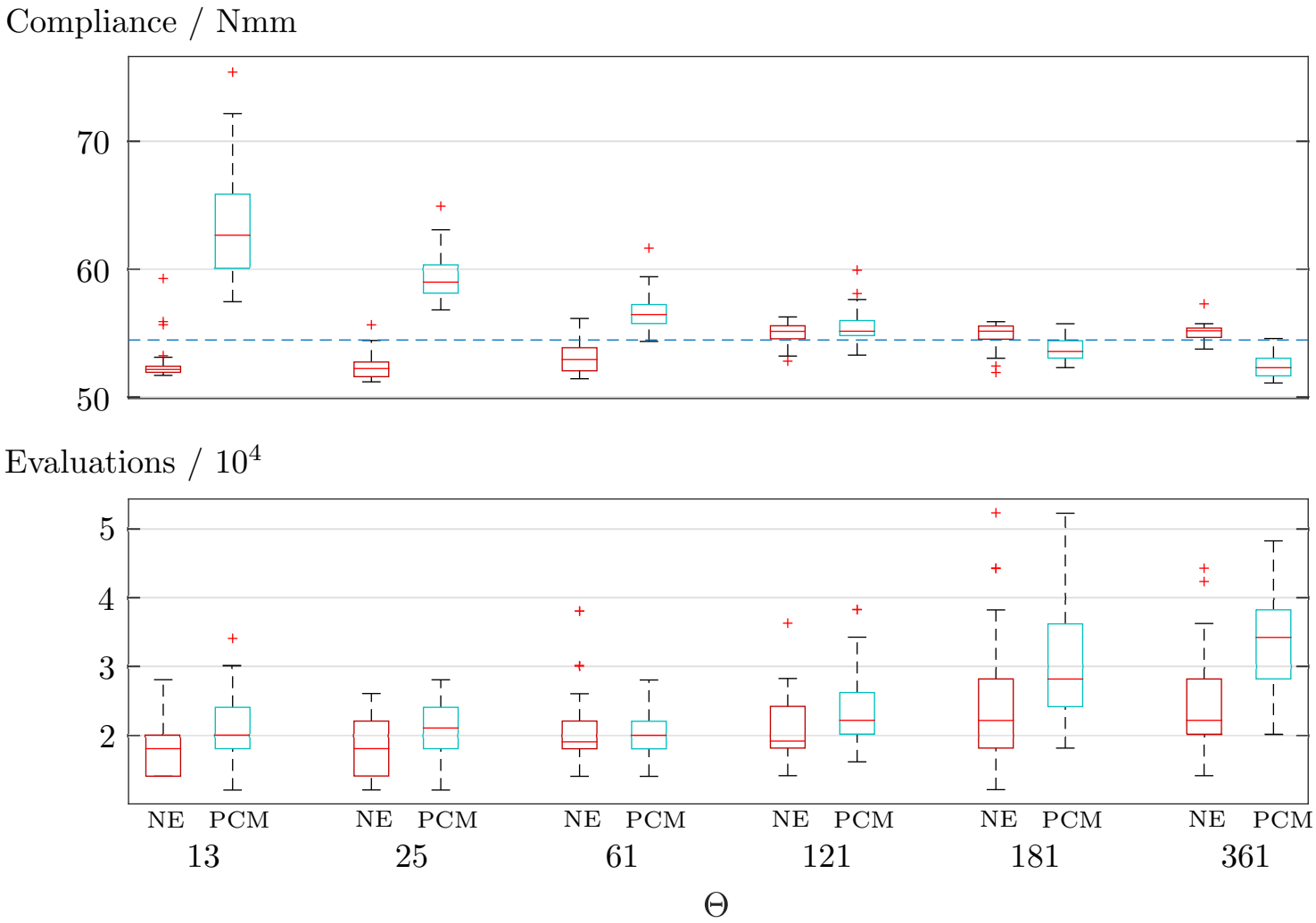


Figure 5.3: Optimized compliance values and required evaluations of the NE/PCM-TOPS runs obtained in the experiments with LSF Vector I for different model complexities. The dashed line shows the compliance of the reference structure.

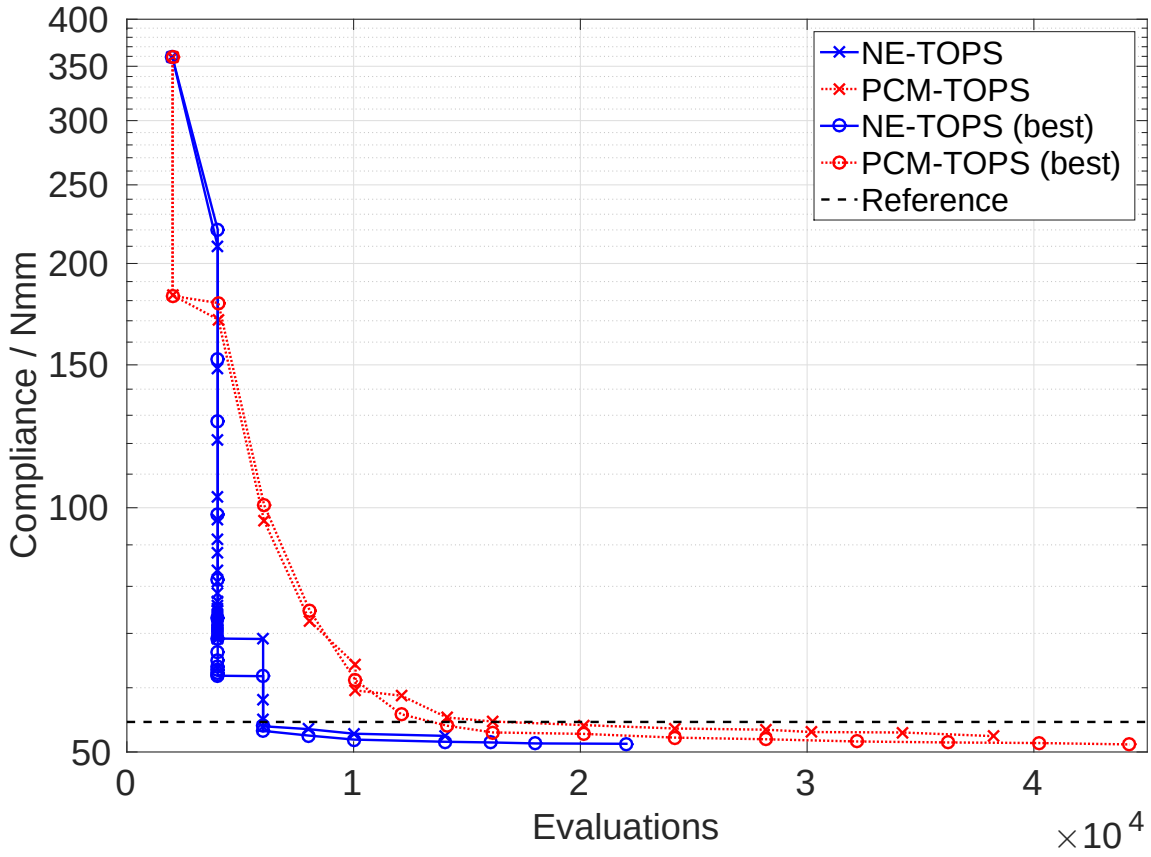


Figure 5.4: The compliance of the mean and best run versus the number of function evaluations for NE/PCM-TOPS on LSF Vector I.

model every iteration.

This confirms, that the update-signal trained in one iteration for one specific structure can be generalized to improve the structure in the next or several following updates. Figure 5.5 shows the evaluations for learning, and the phases of re-usage of the model for the best run. For NE-TOPS, learning is occurring rarely and the model is reused for a long period (iterations without evaluations). In the last iterations, as the structure converges, the learning effort increases, as it becomes harder to find a model that improves the structure. In the last iteration, the maximum number of possible evaluations are utilized and unable to find an update-signal that improves the structure. For PCM-TOPS the update-signal model is only reused once and stopping occurs earlier, yet the same behaviour of increasing evaluations towards the end is observed.

The optimized structures are shown in Fig. 5.6. On average, the resulting structures show good visual similarity to the reference structure.

Figure 5.7 shows the empirical correlation coefficient between the update

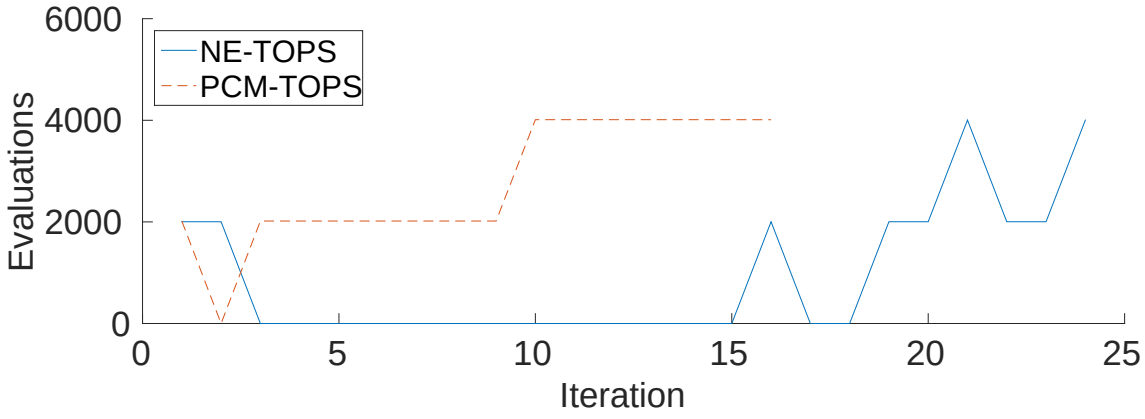
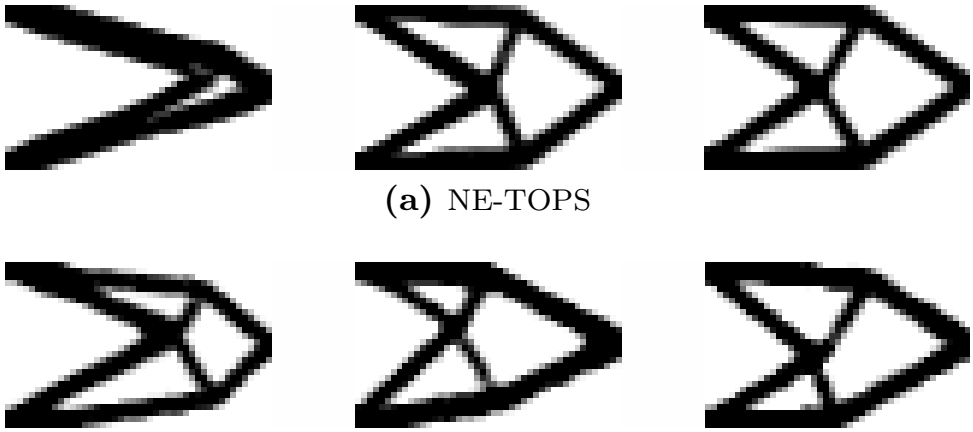


Figure 5.5: The number of function evaluations for NE/PCM-TOPS learning steps, over the iterations of the topology optimization.



(a) NE-TOPS

(b) PCM-TOPS

Figure 5.6: Optimized structures obtained for LSF Vector I with $\Theta = 361$, from left to right: worst run, mean run and best run.

signals and the analytical sensitivities versus the number of iterations. Pearson's correlations coefficient is a measure for the linear correlation between two quantities, where 1 indicates a perfectly linear relation and 0 indicates no linear correlation. More information and its definition is given in App. A.4. The results show that the correlation is always positive for both learning approaches. However, NE-TOPS achieves a much higher and also more flat correlation than PCM-TOPS. Hence, the prediction by NE-TOPS are more linearly related to the actual sensitivity.

The results show that it is possible to obtain the reference structure for both evolutionary learning approaches, although the method is most economic for a simple model with NE-TOPS. The results are obtained without

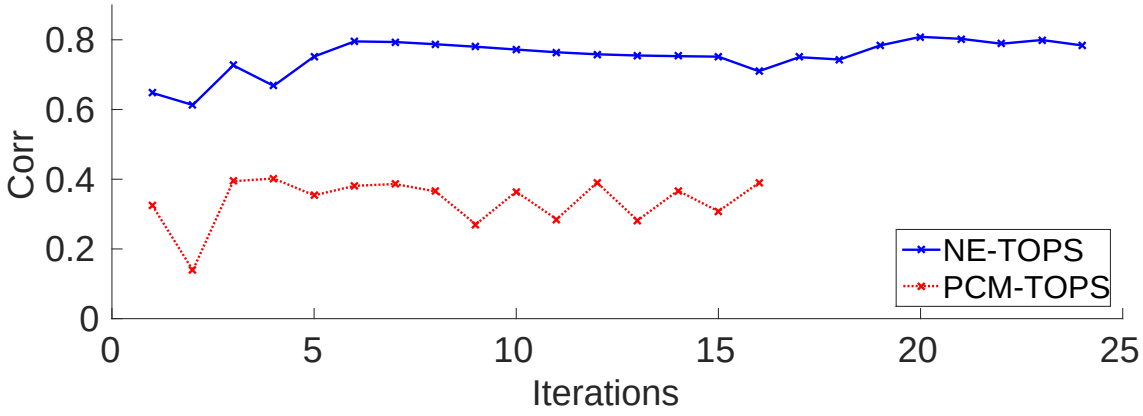


Figure 5.7: Empirical correlation coefficients for update-signals with sensitivities for NE/PCM-TOPS and LSF Vector I.

retraining a new update-signal model every iteration. This demonstrates that in some iterations, the update-signal model is generalized to a new design and instantly performing well. For LSF Vector I, TOPS with evolutionary optimization is capable of successfully substituting the analytical sensitivities by recognizing how to use the information contained in the feature vector.

5.2.2 Results LSF Vector II

This section describes the results for LSF Vector II, where the strain energy density is not included as LSF. An overview of the results in terms of compliance and evaluations is shown in Fig. 5.8. While for PCM-TOPS there is a steady trend of decreasing compliance, for NE-TOPS no statistical significant change in compliance is observed between 10, 15 and 30 hidden neurons. The evaluations increase and then start to stagnate for higher number of model parameters. For both approaches the lowest mean compliance is obtained for the highest model complexities (30 hidden neurons and 331 prototypes). The results are further studied for these model complexities that resulted in the lowest mean compliance.

On average, NE-TOPS stops after $4.4 \cdot 10^4$ evaluations, while for PCM-TOPS, the number of evaluations is notably lower with $3.6 \cdot 10^4$ evaluations. Although it requires less evaluations, PCM-TOPS results in significantly lower mean compliance. Figure 5.9 shows the compliance objective versus the number of function evaluations. Again generalizations of the update-signal model show up as steep descents in some iterations, in which the update signal model can be reused.

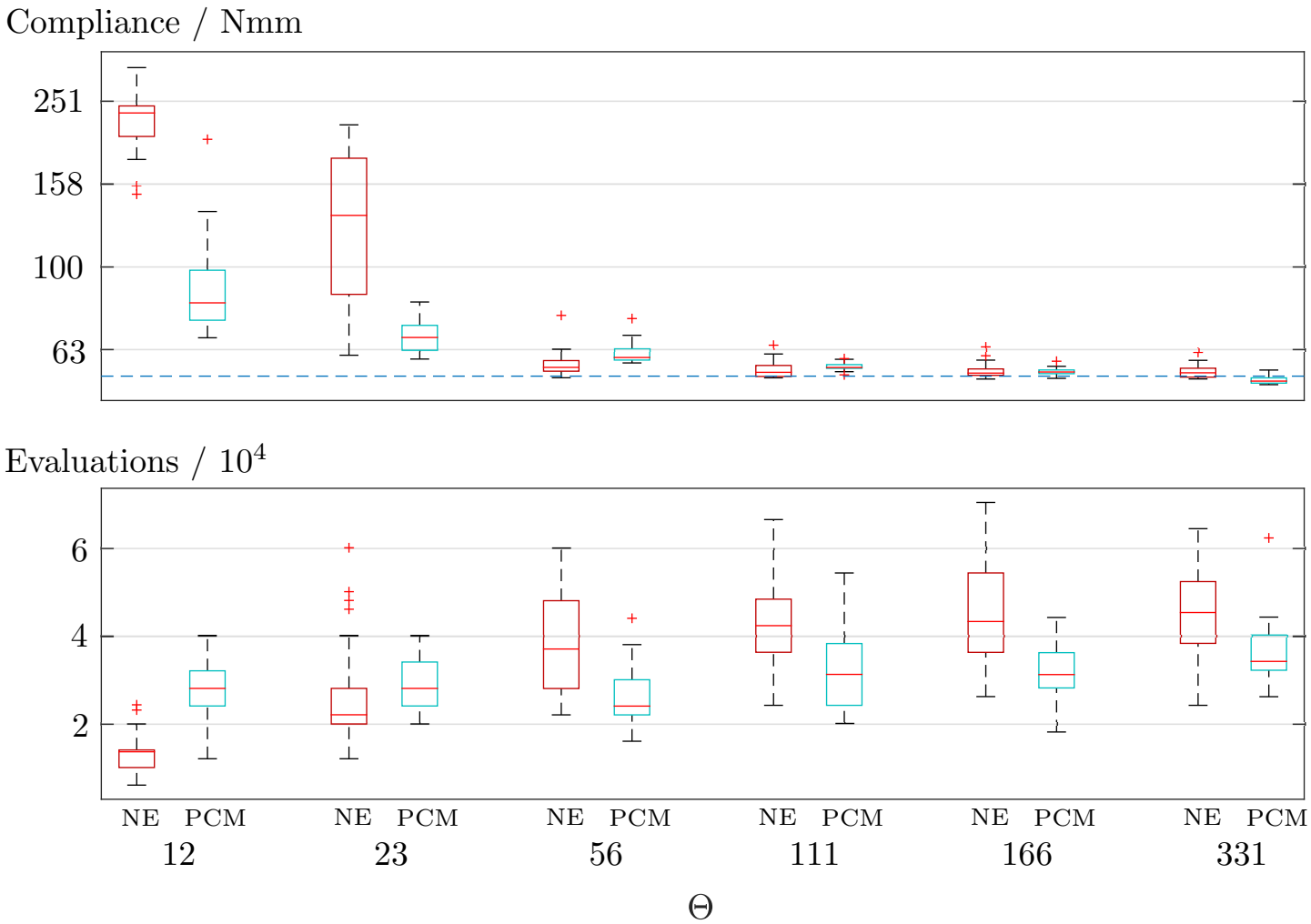


Figure 5.8: Optimized compliance values and required evaluations of the NE/PCM-TOPS runs obtained in the experiments with LSF Vector II. The dashed line shows the compliance of the reference structure.

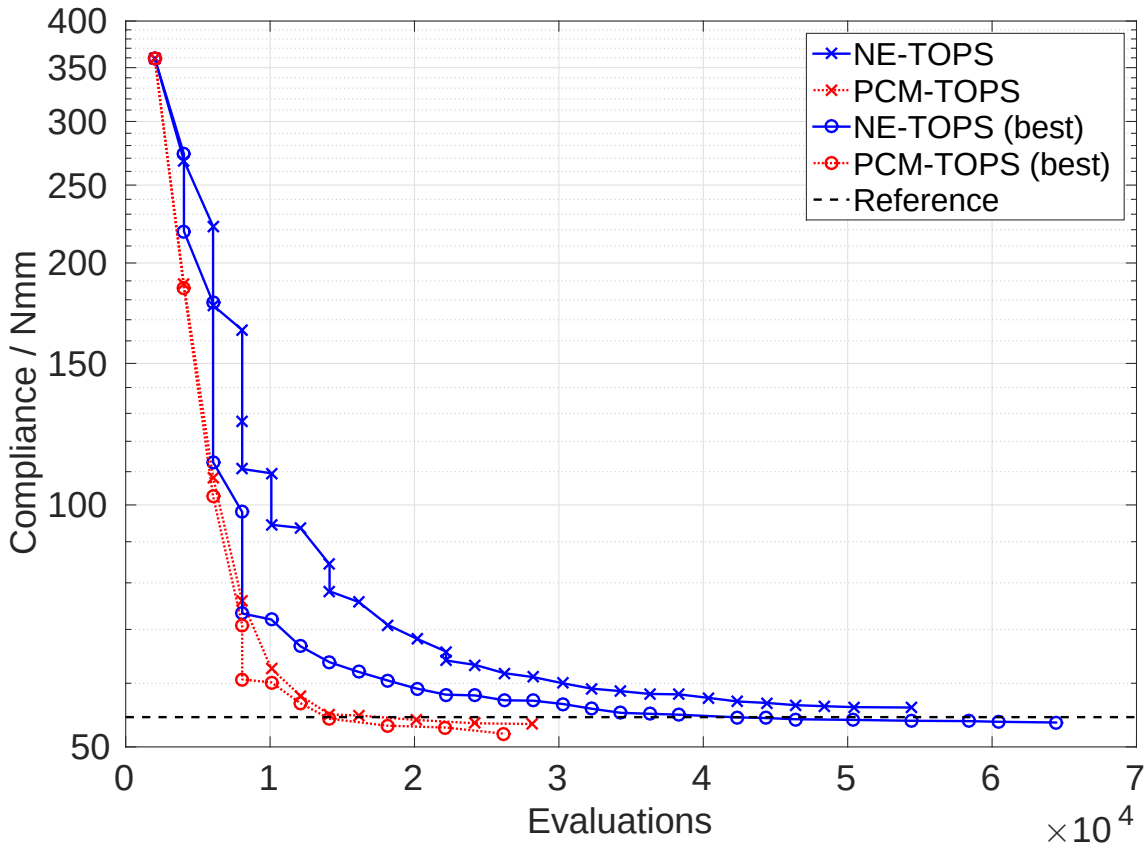


Figure 5.9: The compliance of the mean and best run versus the number of function evaluations for NE/PCM-TOPS on LSF Vector II.

For the best run the evaluations used for update-signal model learning are shown in Fig. 5.10. NE-TOPS reuses the update-signal model in several of the iterations, PCM-TOPS in one iteration.

The resulting structures are shown in Fig. 5.11. The reference structure is reproduced by PCM-TOPS, but for NE-TOPS even the best structure lacks the connections of the reference cantilever and only the simple two beam cantilever is obtained.

Figure 5.12 shows the empirical linear correlation coefficient between the update signals and the analytical sensitivities versus the number of iterations (More information and the coefficient's definition are given in App. A.4). At least a weak correlation in all iterations is observed. In the later iterations NE-TOPS achieves a higher correlation than PCM-TOPS. This demonstrates that PCM-TOPS achieves the reference result, although it is not clearly modelling the actual sensitivities, but instead alternative update-signals to improve the structure. Nevertheless this leads to structures that more closely represent the reference than NE-TOPS.

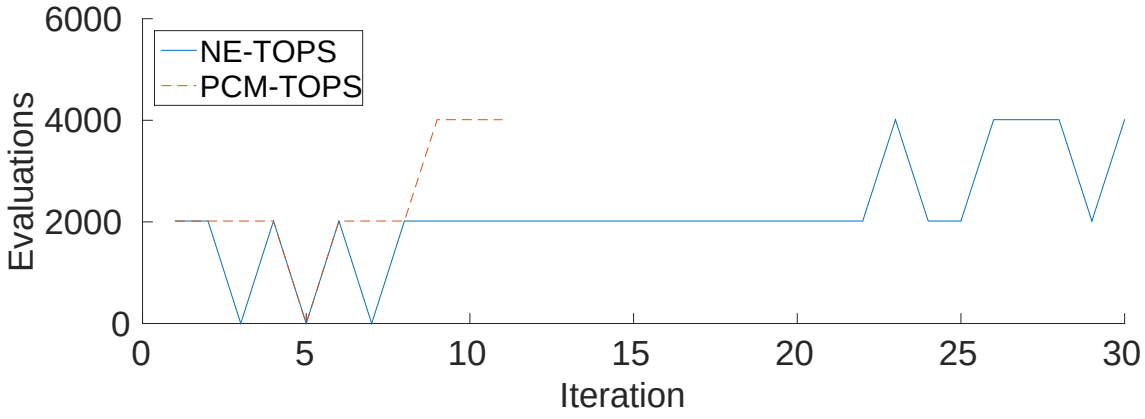
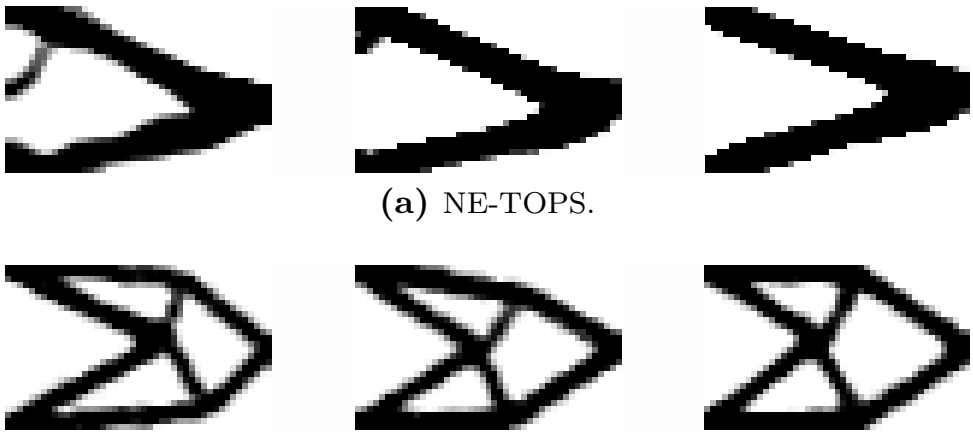


Figure 5.10: The number of function evaluations for NE/PCM-TOPS learning steps, over the iterations of the topology optimization.



(a) NE-TOPS.

(b) PCM-TOPS.

Figure 5.11: Optimized structures obtained for NE-TOPS (top row) and PCM-TOPS (bottom row) for LSF Vector II with $\Theta = 331$, from left to right: worst run, mean run and best run.

5.2.3 NE/PCM-TOPS Results Discussion

A large difference in the performance of the different modelling approaches with the two LSF vectors can be observed.

NE-TOPS with LSF Vector I performs better compared to LSF Vector II. This can be seen by lower compliance values that are obtained, lower variance of the resulting designs, and a reproduction of the reference structure. Importantly, these better results are obtained with a much lower amount of function evaluations. The superiority is founded in the usage of the strain energy density as a LSF. As discussed in Sec. 5.1, the strain energy density of an element is proportional to the analytical sensitivity.

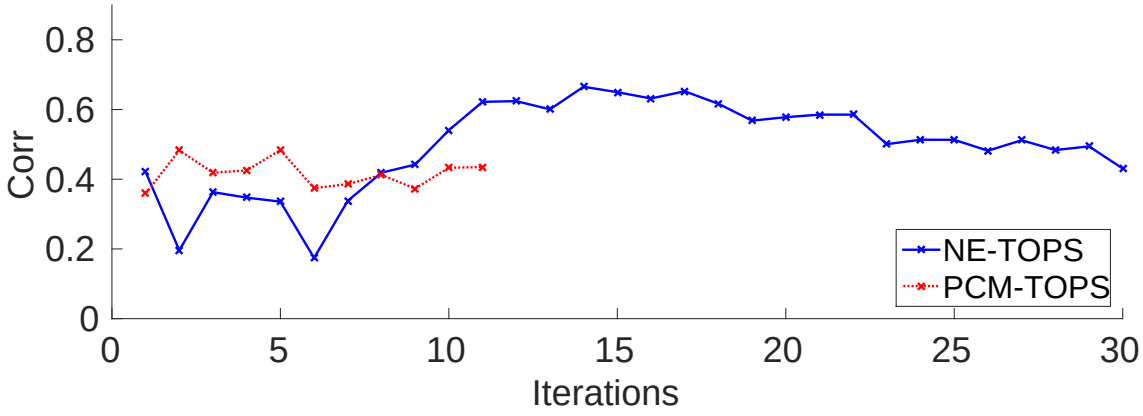


Figure 5.12: Empirical correlation coefficients for update-signal with sensitivities and LSF Vector II.

Thus, the learning task for the neural network is simple, since basically a linear relation needs to be realized. This is easily possible by choosing the weight parameters accordingly, such that the output mainly reflects the strain energy density LSF, therefore, an improvement is expected for its usage.

This indicates that the NE-TOPS approach offers the possibility to select the appropriate LSFs. From Fig. 5.3, it can be seen that the mean compliance is even slightly better than that of the reference. It is achieved not by close modelling of the sensitivities, but by optimizing the update-signals such that the number of elements with intermediate densities is reduced. The improvement itself is not relevant, since the same concept structure is obtained. However, it reminds us that due to the generic nature of the approach, it would be possible to include a post-processing step that generates a discrete solution directly into the evaluation step.

PCM-TOPS requires a much higher number of model parameters than NE-TOPS for LSF Vector I. This lies in the nature of the PCM modelling approach that distributes prototypes within the 10-dimensional LSF space and has no means of preferring a specific LSF. Hence, the prototypes are distributed sparsely and it is unlikely that the prototypes are distributed along the dimension of the strain energy density, which renders the achievable linear relation a question of chance. When the sparseness is reduced, by reducing the number of LSFs or by increasing the number of prototypes, PCM-TOPS can improve in competitiveness. Figure 5.13 shows the significant reduction in compliance and evaluations, when a reduced LSF Vector I with only density and strain energy density is applied for the case of $P = 13$. Hence, PCM-TOPS might benefit substantially from a

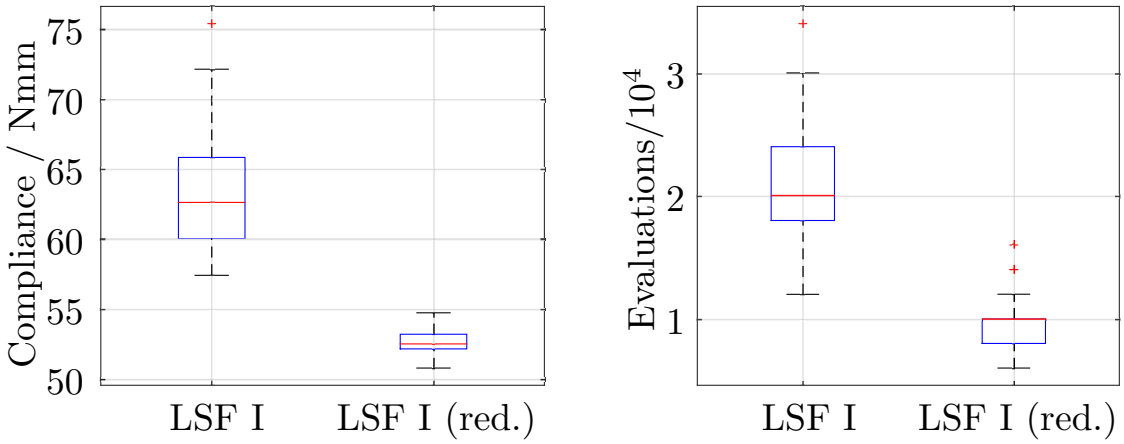


Figure 5.13: Comparison of the resulting compliance (left) and the required evaluations (right) of PCM-TOPS with $P = 13$ for LSF Vector I and a reduced LSF Vector I. In the latter, only density and strain energy density are present, without nodal displacements. For the small number of prototypes, PCM-TOPS can profit significantly from a reduced LSF space dimensionality and a much lower compliance is obtained along with lower computational cost.

systematic feature selection or dimensionality reduction technique. As alternative the number of prototypes can be increased to improve the result, yet more evaluations are required (see Fig. 5.5).

However, the strain energy density is not of fundamental importance for PCM-TOPS. The approach improves for LSF Vector II compared to LSF Vector I. The higher the number of prototypes, and the lower the number of LSFs, the more the quantization error can be reduced. Therefore, the reference can be obtained reliably for PCM-TOPS with LSF Vector II. This is not the case for the neural network model, albeit its compliance values are almost as good as the reference. These are obtained by fine-tuning the densities towards a more discrete solution, yet it fails to reproduce the components of the reference structure, which is the desired target. For NE-TOPS the lack of the strain energy density in LSF Vector II implies a more challenging learning task, that can be seen by higher solution variances and the higher number of required function evaluations.

The results for LSF Vector II are especially interesting from the generalization point of view. Since a strong feature like the strain energy density might not be available for other objective functions, it is important that it is possible to generically substitute the non-linear relation of basic features and the analytical sensitivity.

The results suggest that the optimization of a small neural network to

select from one of the LSFs is computationally more efficient compared to the PCM. PCM-TOPS is a better choice when the relation of the LSFs to the sensitivity is non-linear. It does not depend as much on the choice of LSFs, under the assumption of a sufficiently high number of prototypes.

Figures 5.7 and 5.12 show that the learned update-signals correlate weakly and moderately with the sensitivities. Apparently, also update-signals with weak correlation can improve the structure. In these cases, the model should rather be considered as an indirect structure representation for EC than a sensitivity model.

By reproduction of the reference structure as well as the reference objective values, the generic approach is shown to be feasible, even when the utilized LSFs contain the sensitivity information only implicitly. The results demonstrate that in all experiments the re-usage of the update-signal models reduces the number of evaluations, since the model is not re-trained in every iteration, and the optimization does only require the limit of evaluations $M^{\max}$ towards the end of the optimization.

However, both approaches require a large number of evaluations, and from a practical perspective a further reduction would be desirable. One possibility is the explicit learning of the sensitivity function by finite difference sampling. This is the focus of the next section.

5.3 SVR/LIN-TOPS Experiments

Several experiments are conducted to perform an empiric validation of the generic topology optimization approach based on supervised learning. The experiments consider the approaches introduced in Sec. 4.4, namely:

- Linear Regression Topology Optimization by Predicting Sensitivities (LIN-TOPS),
- Support Vector Regression Topology Optimization by Predicting Sensitivities (SVR-TOPS),
- Support Vector Regression Topology Optimization by Predicting Aggregated Sensitivities (SVR-TOPAS).

LIN-TOPS and SVR-TOPS are evaluated for both LSF vectors introduced in Sec. 5.1.1. Computational cost in terms of structure evaluations are lower compared to the explicit evolutionary learning. This facilitates additional experiments considering various other LSF vectors, a mesh-dependency study, and a modified objective function subject to noise. For

Table 5.3: Parameter settings for LIN/SVR-TOPS experiments.

Setting	Symbol	Value
FD step size	ϵ	0.01
Number of remembered learning events	-	8
Evaluations per learning step	M	100
Max. evaluations per iteration	$M^{\max}$	500

the latter, SVR-TOPAS is evaluated, since it achieves a very good reproduction of the reference and does not require additional features. The parameter specifications of the experiments are given in Tab. 5.3.

The linear regression is considered in order to test the hypothesis of linear relations between LSF and the sensitivity. Naturally, the linear regression model (4.13) cannot model the sensitivity (5.1) well when only LSF Vector II is considered. However, it can represent a multivariate polynomial in LSF Vector II with higher degree when additional features are constructed by multiplication of the appropriate components. We designate $\mathbf{s}_i^{\mathrm{II}_{\mathrm{2nd}}}$, $\mathbf{s}_i^{\mathrm{II}_{\mathrm{3rd}}}$, $\mathbf{s}_i^{\mathrm{II}_{\mathrm{4th}}}$ as the LSF vectors that additionally contain all possible 2nd, 3rd and 4th order terms in the components of LSF Vector II. The high number of features resulting from these constructions in combination with relatively little available training data renders the regularization of the linear regression especially important. As described in App. A.2, over-fitting is avoided by the quadratic regularization term added to the ordinary least squares problem of linear regression. The regularization parameter is found by a line search on a cross validation data set.

Besides the linear regression, the support vector regression as a state-of-the-art non-linear predictor is applied (see Sec. 4.4.3). Hus et al.ś recommendations for the SVR are followed, i.e. the RBF kernel is used and the important parameters are found by a grid search on a subset of the training samples [87]. The SVR is intrinsically able to model non-linear relations. Therefore, it is not applied in combination with constructed higher order features, but instead with other potential features like elemental stresses or strains.

For all experiments, forward finite differences are used. Samples of the last eight designs, which have been sampled, are remembered, older samples are neglected for training. All experiments were run for 30 different random seeds, in order to account for the stochastic nature of the sampling

choice. Quantitative results for all experiments are collected in App. B. Conclusions are based on statistical t-test that are performed based on the 5% significance level.

The implementation is the same as described in Sec. 5.2, using the LIBSVM[3] [41] software library for the SVR. See also the reference for the specific formulation of the ϵ-SVR optimization problem that is solved.

The next subsections briefly describe the results for the two predictors using the two LSF vectors in Sec. 5.3.1 and 5.3.2, followed by an intermediate discussion in Sec. 5.3.3. An overview of the results is given in Fig. 5.14. Then, experiments with varying mesh-sizes, alternative LSF, and experiments with aggregated sampling are presented in Sec. 5.3.4 to 5.3.6, respectively.

5.3.1 Results LSF Vector I

From Fig. 5.14, we see that using LSF Vector I, which includes the strain energy density, yields low compliance values at low evaluation cost. The designs resulting from LIN-TOPS and SVR-TOPS with LSF Vector I are depicted in Fig. 5.15. From visual comparison it can be seen that the baseline structure is reproduced closely.

This is also reflected in the compliance values. The lowest compliance in the whole study is achieved when using LSF Vector I combination with linear regression. The linear regression directly enables a sensitivity model proportional to the strain energy density and even achieves a lower compliance than the reference. The optimized compliance of the support vector regression is slightly higher than the reference, but this is not visible in the resulting structure.

For both models, training requires few evaluations, since the linear relation between strain energy density and sensitivity can be learned easily from few sample points. On average, about 5.8% of the reference evaluations M_{ref} are required. Figure 5.16 shows that the retraining occurs sparsely, and only utilizes the minimum amount of samples except for the final iteration.

5.3.2 Results LSF Vector II

Using LSF Vector II, a non-linear relation between LSF and sensitivity has to be modelled that can hardly be approximated by a linear regression

[3] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html`

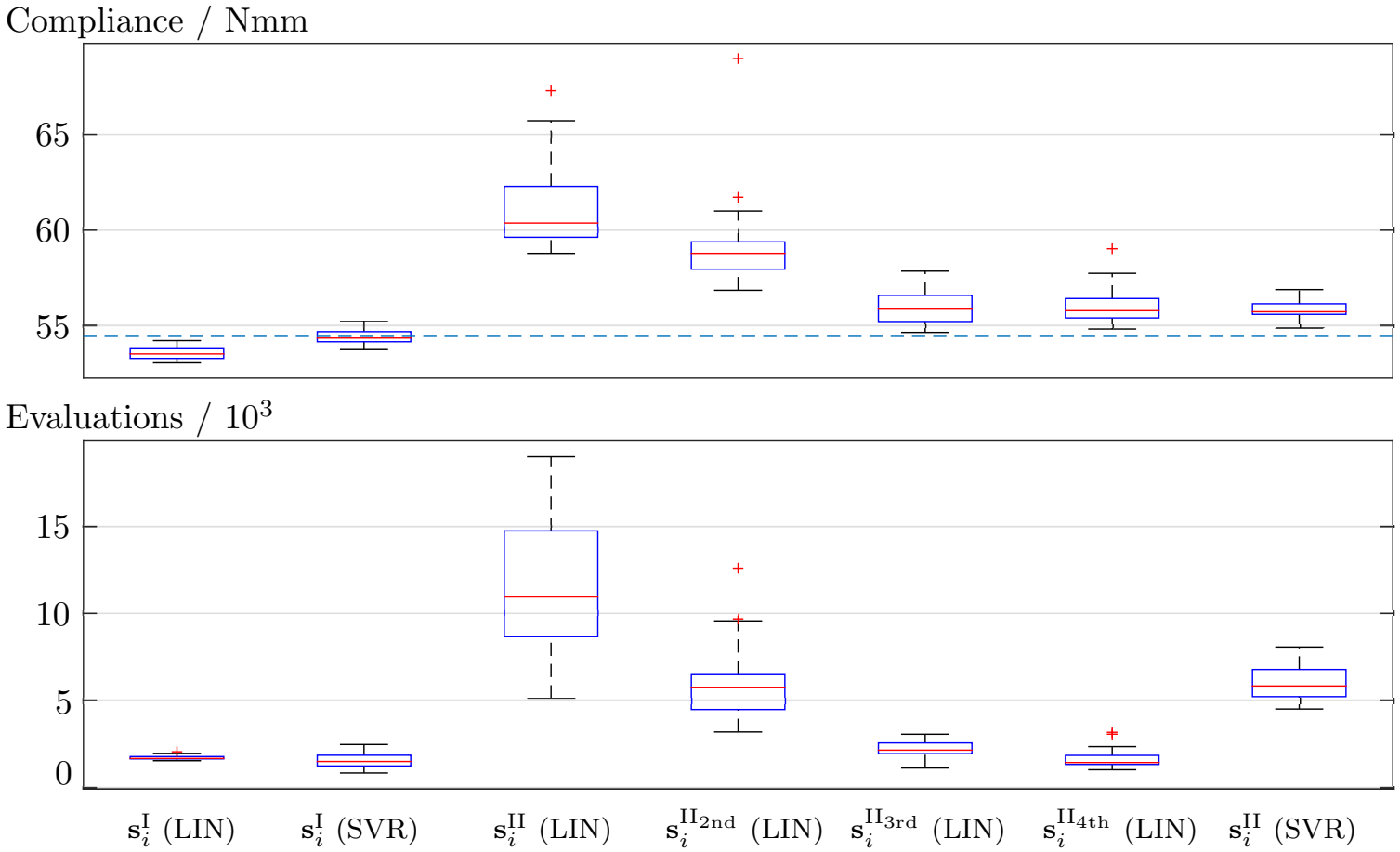
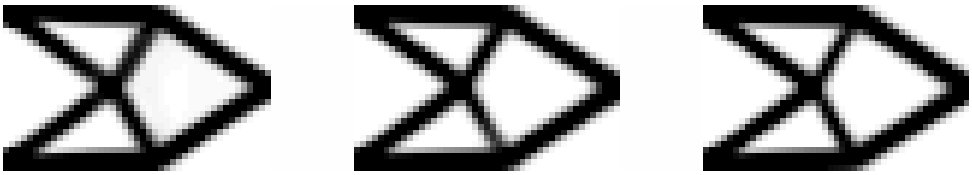
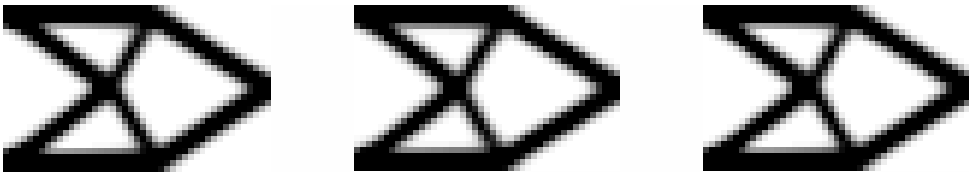


Figure 5.14: Optimized compliance values and required evaluations of the runs obtained in the different experiments with LIN/SVR-TOPS and LSF Vectors I and II. For linear regression higher order terms of the vector components of LSF Vector II are considered. The dashed line shows the compliance of the reference structure.



(a) LIN-TOPS, $\mathbf{s}_i^{\mathrm{I}}$.



(b) SVR-TOPS, $\mathbf{s}_i^{\mathrm{I}}$.

Figure 5.15: Optimized structures obtained for LIN-TOPS and SVR-TOPS for LSF Vector I; from left to right: worst run, mean run and best run.

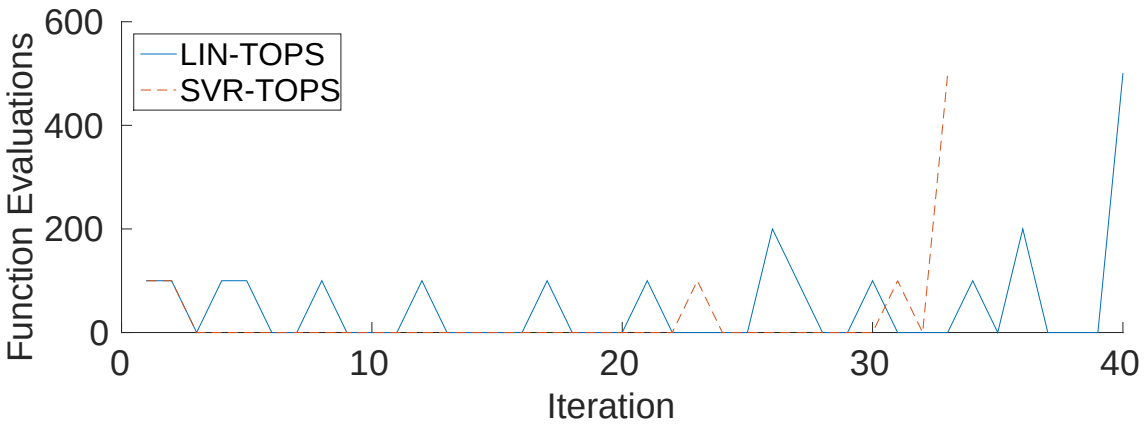
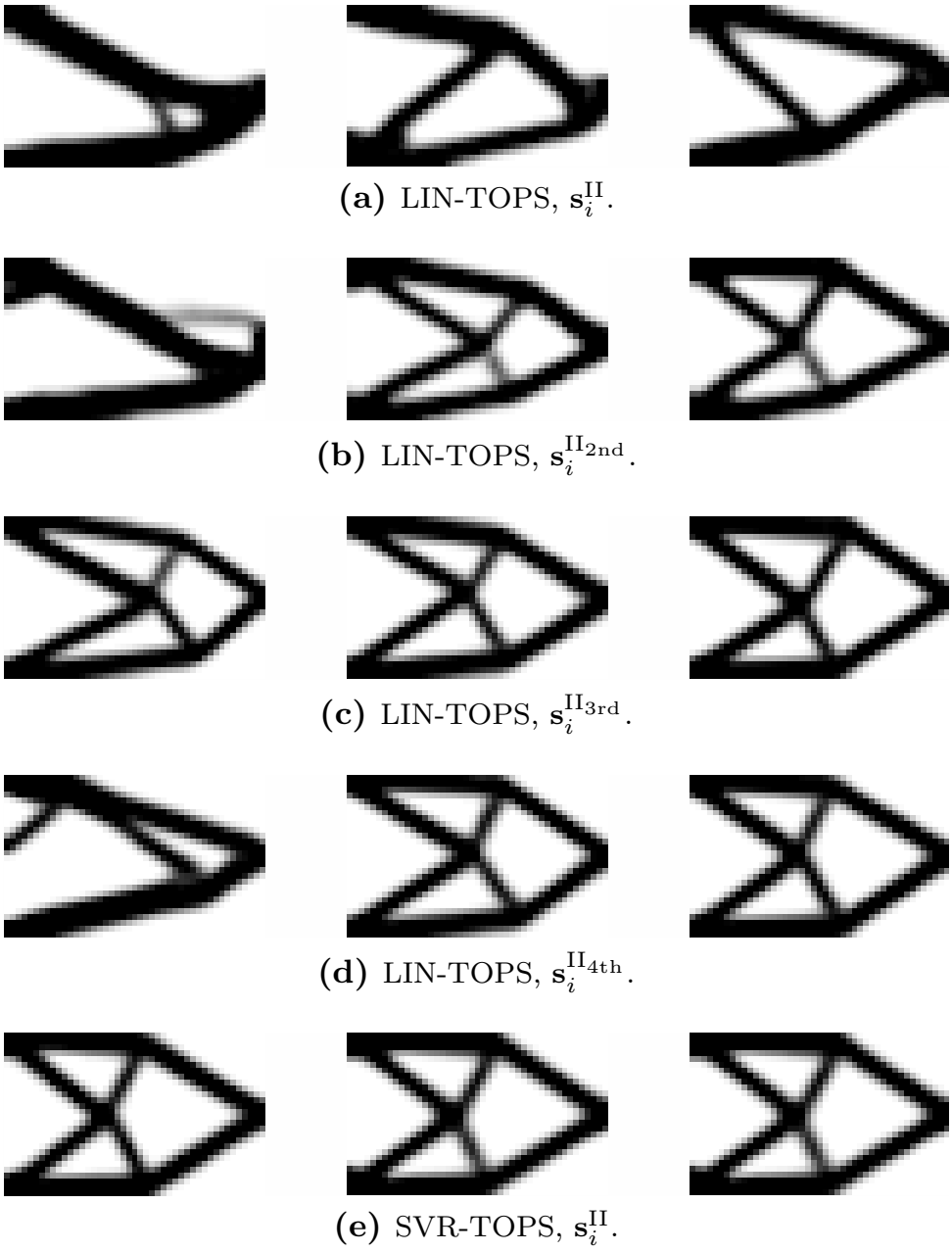


Figure 5.16: The number function evaluations of LIN/SVR-TOPS required for learning, over the iterations of the topology optimization for LSF Vector I.

model. This can be concluded from the compliance value obtained, which is higher than that of the reference structure. It can also be concluded from the resulting structures in the top row of Fig. 5.17 that involve variance and lacks one or several of the connections of the reference structure. Even this result is only achieved with a relatively high number of evaluations for learning (about 10,000), due to the fact that the linear model can only achieve rough local representations of the sensitivity. It needs frequent retraining as can be seen in Fig. 5.18, exploiting the range in between a minimum of 100 and the maximum of 500 evaluations.

The advantage of the SVR in this scenario is its ability to learn the required non-linear relation between the LSFs and the sensitivity. This



(a) LIN-TOPS, $\mathbf{s}_i^{\mathrm{II}}$.

(b) LIN-TOPS, $\mathbf{s}_i^{\mathrm{II}_{2\mathrm{nd}}}$.

(c) LIN-TOPS, $\mathbf{s}_i^{\mathrm{II}_{3\mathrm{rd}}}$.

(d) LIN-TOPS, $\mathbf{s}_i^{\mathrm{II}_{4\mathrm{th}}}$.

(e) SVR-TOPS, $\mathbf{s}_i^{\mathrm{II}}$.

Figure 5.17: Optimized structures obtained for LIN-TOPS and SVR-TOPS for LSF Vector II from left to right: worst run, mean run and best run.

results in the accurate reproduction of the reference structure, shown in the bottom line of Fig. 5.17. Although retraining occurs not as frequent as for the linear regression, about roughly 6,000 evaluations are required, which reflects the more challenging learning task. Still, this is only about 23% of the reference approach evaluations M_{ref} based on pure finite differentiation.

For linear regression, a trend of reducing compliance is observed when using higher order variants of LSF Vector II. Adding second, third and

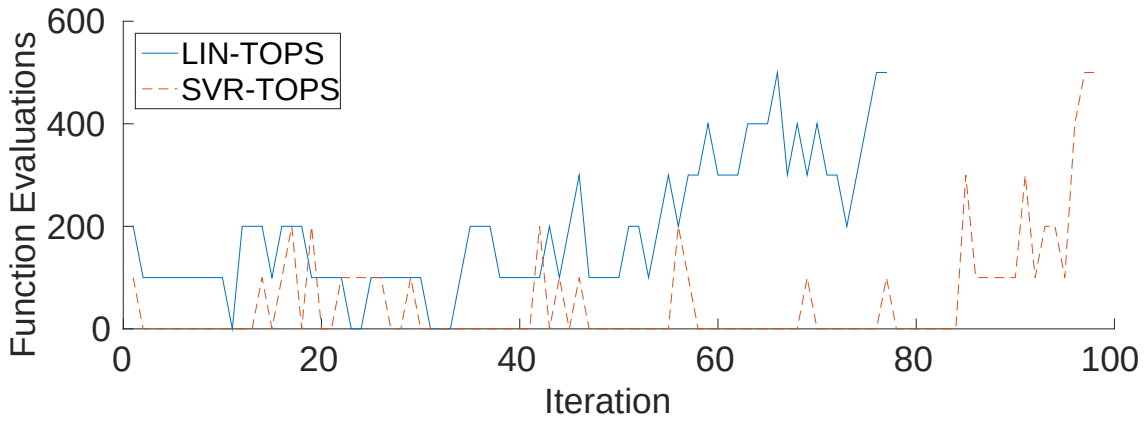


Figure 5.18: The number function evaluations of LIN/SVR-TOPS required for learning, over the iterations of the topology optimization for LSF Vector II.

forth order terms of the components of LSF Vector II not only improves the compliance but also reduces the required number of function evaluations. The resulting structures are shown in Fig. 5.17 in the three central rows. The higher the order of LSF Vector II components, the better also the visual resemblance between the optimized design and the reference. On average, the reference structure is reproduced for $\mathbf{s}_i^{\mathrm{II}_{3rd}}$ and $\mathbf{s}_i^{\mathrm{II}_{4th}}$. This is also confirmed by the fact that there is no statistically significant difference between these runs and SVR-TOPS with LSF Vector II. The advantage of using higher order LSF can be explained by the analytical formulation of the sensitivity (5.1) which is a multi-variate polynomial. Thus, the construction of higher order LSF for the linear regression results in a multi-variate polynomial model in the components of LSF Vector II. With increasing feature order, the model is more and more able to approximate the sensitivity function. This improves the result and reduces the learning effort.

Compared to the reference, only 6% of the function evaluations are required when using $\mathbf{s}_i^{\mathrm{II}_{4th}}$. This is the lowest number of evaluations for the linear regression predictor that is achieved in the study with LSF Vector II.

5.3.3 Intermediate LIN/SVR-TOPS Results Discussion

In most experiments, optimized designs similar to the reference design are obtained, while the number of finite element solver runs is reduced

drastically, compared to naive finite differentiation gradient estimation. The solution quality and the number of required samples depend on the prediction quality. Recommendations for the choice of update-signal model and features can be drawn based on the conducted experiments.

Results can be interpreted when considering the analytic formulation of the sensitivities given in (5.1). As for the results of NE/PCM-TOPS in Sec. 5.2, considering the strain energy density has a high impact on the results. Its usage results in significantly low compliance for the linear regression as well as a strong reduction of evaluations for the SVR.

When considering the results of SVR-TOPS in terms of compliance, both feature sets perform similar. This is also reflected in the visual resemblance of the optimized designs to the reference. Although the strain energy density contains much more information on the sensitivity, the same information is also contained implicitly in LSF Vector I. The SVR can model the non-linear function, however profits from a significantly lower learning effort in case the strain energy density is available a-priori.

For LSF Vector II, SVR-TOPS yields a solution very similar to the reference, with 77% less function evaluations compared to the baseline, since it is able to learn the suitable non-linear relation. The roughly 6,000 FEA simulation in this case seem still to be a high computational cost. However, since the sensitivity function is the same for all elements, it is intuitive to expect that the absolute number of samples required for learning will not rise with an increased mesh size, a hypothesis that is validated in the next section.

The linear regression cannot model the non-linear relation well, unless higher order LSF are constructed. The linear regression potentially requires less samples for learning, depending on how well the sensitivity function can be modelled by a polynomial. In fact there is no statistical difference between the number of evaluations required for LSF vectors $\mathbf{s}_i^{\mathrm{I}}$ or $\mathbf{s}_i^{\mathrm{II}_{4\mathrm{th}}}$. Together with SVR-TOPS on LSF Vector I, these runs are the overall cheapest, reducing the number of evaluations to about 6% of the reference.

With both predictors, it is possible to obtain good results. SVR produces good results for both LSF vectors while linear regression relies on feature construction. However, SVR comes with the drawback of a higher computational cost for training compared to linear regression. The lowest compliance is achieved for linear regression with LSF Vector I. The smallest amount of samples is required in case of the Linear Regression with $\mathbf{s}_i^{\mathrm{II}_{4\mathrm{th}}}$. The SVR optimization result is robust to feature choice, however it requires different amount of samples. This leads to the question: Which

impact do other LSF vectors have? This is studied in Sec. 5.3.5.

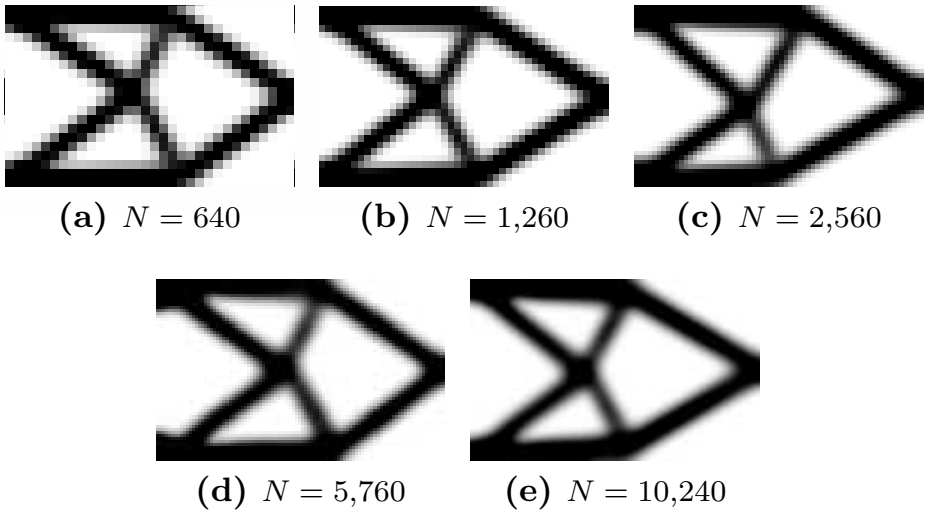
The results suggest that in general it is advisable to use a non-linear predictor like SVR and to include all LSF that might be relevant. If expert knowledge or intuition on the type of non-linearity is available, this can be useful to chose or construct additional LSF, like higher order versions. In this case, a linear regression can be more efficient than SVR.

5.3.4 Mesh-Independency Study

The experiments in the previous section demonstrate that the number of evaluations can be notably reduced compared to a naive finite differentiation approach. This capability becomes even more important when the number of design variables in the reference design space is increased.

We rerun SVR-TOPS with LSF Vector II for different mesh sizes: $N \in \{32 \times 20 = 640, 64 \times 40 = 2{,}560, 96 \times 60 = 5{,}760, 128 \times 80 = 10{,}240\}$. Figure 5.19 shows the best structure of the experiments. The reference cantilever structure is reproduced in all cases.

More interestingly, the average number of evaluations is stagnating independently of the mesh size, as is visible in Fig. 5.20. For $N = 1{,}260$, SVR-TOPS requires 23% of M_{ref}. For a mesh size of $N = 10{,}240$ (when we assume the same number of iterations for the naive finite differentiation approach), the fraction required by SVR-TOPS reduces to roughly 2%. We can conclude that the number of evaluations is mesh-independent for



(a) $N = 640$ **(b)** $N = 1{,}260$ **(c)** $N = 2{,}560$

(d) $N = 5{,}760$ **(e)** $N = 10{,}240$

Figure 5.19: Best optimized structures of SVR-TOPS with LSF Vector II for different mesh sizes.

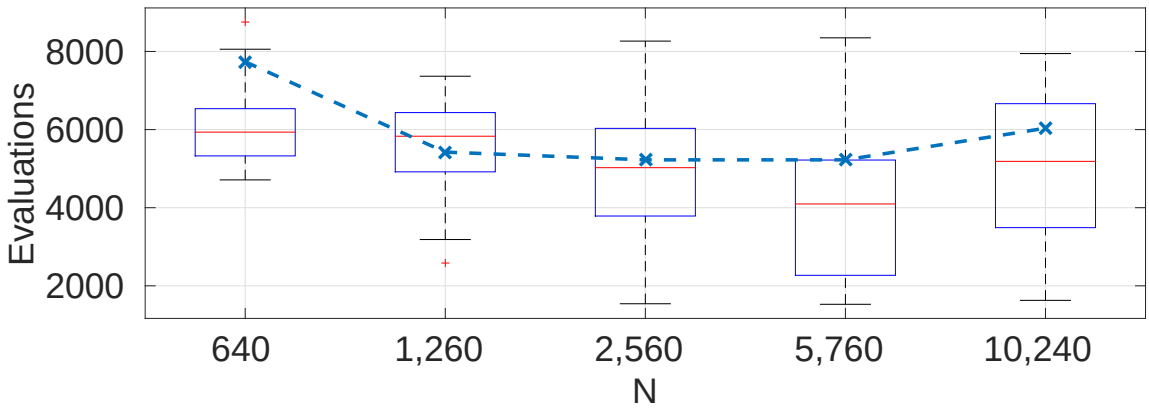


Figure 5.20: Required evaluations of SVR-TOPS with LSF Vector II for different mesh sizes. The dashed line shows the evaluations required by the run that resulted in the best structure.

SVR-TOPS in this setting.

A problem that arises with larger mesh sizes is that the random choice of sensitivity samples can be increasingly unrepresentative of the actual distribution. Possibly, this causes some runs to stop early. These have to be considered as failures, yet they enter the statistics in Fig. 5.20 with only few evaluations, especially for higher mesh sizes. Therefore, the evaluation numbers of the best runs are added to the plot (dashed line). This confirms that for the best optimized structures a significant increase in evaluations is *not* necessary when increasing the mesh size. In the future, techniques are required that choose the most appropriate samples in a deterministic way, for instance based on confidence measures.

Another observation is that the obtained structures themselves achieve mesh-independency, at least for the considered reference structure. As discussed in Sec. 2.2.2 a finer mesh enables to represent a finer structure with more details, causing mesh dependence of the optimum structure. The mesh-independency of the designs is achieved by the included filtering of sensitivities approach [166, 169], see (4.6). The filter radius is constant for all mesh sizes and, hence, the same minimum length scale is defined. This principle achieves mesh-independency for general boundary conditions. Hence, assuming the sensitivity prediction is adequately precise, we can expect the mesh-independency also for different load and support conditions and even other objective functions. Note also the comment on generality of the results in Sec. refsec:compoveraldiscussion.

5.3.5 Alternative LSF

With the LSF Vector I and LSF Vector II two extreme cases were considered. LSF Vector I represents the case of almost analytical sensitivity information within the LSF vector. LSF Vector II represents minimum previous knowledge, since the nodal displacements are the most basic data obtained from a finite element simulation. In application of TOPS, the user needs to select LSFs using available expertise. Problems, for which only displacement information is available might be rare, as well as problems, for which a LSF correlates strongly with the sensitivity. Thus far, only the nodal displacement, the strain energy density and the density are used in the LSF vectors. In this section, the idea of using alternative physical variables and coordinate information as LSF is explored using SVR-TOPS.

Physical variables like strain and stress are interesting to evaluate as LSF, since they do not simply relate proportional to the sensitivity (like the strain energy density) but still contain more elaborate information about the state of elements than the displacement. Accordingly, it is expected that including stresses and strains contributes positively to the learning task when added to LSF Vector I.

Although independent of the state and therefore not strictly LSF, further interesting model inputs are the Cartesian coordinates of an element, as they have been used by [45, 62]. When used alone, the coordinates imply that sensitivity is learned only related to the position of the element in the mesh. Coordinate information can validate that the state information is actually of higher use for the sensitivity prediction. It can potentially serve as additional information. As additional LSFs, the average displacements in both coordinate directions of the elemental nodes are considered. Similar to the strain energy density, stresses or strains, these LSFs implicitly process the information on the displacements.

The following LSF Vectors are considered:

$$\mathbf{s}_i^{\mathrm{III}} = \begin{pmatrix} \mathbf{s}_i^{\mathrm{II}} \\ \epsilon_{i1} \\ \epsilon_{i2} \\ \epsilon_{i12} \end{pmatrix}, \ \mathbf{s}_i^{\mathrm{IV}} = \begin{pmatrix} \mathbf{s}_i^{\mathrm{II}} \\ \sigma_{i1} \\ \sigma_{i2} \\ \sigma_{i12} \end{pmatrix}, \ \mathbf{s}_i^{\mathrm{V}} = \begin{pmatrix} \mathbf{s}_i^{\mathrm{II}} \\ x_{1i} \\ x_{2i} \end{pmatrix}, \ \mathbf{s}_i^{\mathrm{VI}} = \begin{pmatrix} \mathbf{s}_i^{\mathrm{II}} \\ \bar{u}_1 \\ \bar{u}_2 \end{pmatrix},$$

$$\mathbf{s}_i^{\mathrm{VII}} = \begin{pmatrix} \mathbf{s}_i^{\mathrm{III}} \\ x_{1i} \\ x_{2i} \end{pmatrix}, \ \mathbf{s}_i^{\mathrm{VIII}} = \begin{pmatrix} \mathbf{s}_i^{\mathrm{VII}} \\ \bar{u}_1 \\ \bar{u}_2 \end{pmatrix}, \mathbf{s}_i^{\mathrm{IX}} = \begin{pmatrix} x_{1i} \\ x_{2i} \end{pmatrix},$$

with the Cartesian elemental coordinates x_{1i}, x_{2i}, the elemental strains

$[\epsilon_{i1}\ \epsilon_{i2}\ \epsilon_{i12}]^{\mathrm{T}} = \boldsymbol{\epsilon} = \mathbf{B}\mathbf{u}_i$, the elemental stresses $[\sigma_{i1}\ \sigma_{i2}\ \sigma_{i12}]^{\mathrm{T}} = \boldsymbol{\epsilon}^{\mathrm{T}}\mathbf{E}$ and the average nodal displacements $\bar{u}_1 = 0.25(u_{i1} + u_{i3} + u_{i5} + u_{i7})$, $\bar{u}_2 = 0.25(u_{i2} + u_{i4} + u_{i6} + u_{i8})$. The matrices $\mathbf{B}$ and $\mathbf{E}$ are required to obtain the stresses and strains from the displacements are given in Appendix A.3.

The number of evaluations and the achieved compliance values are shown in Fig. 5.21 and the optimized structures are shown in Fig. 5.22. In the majority of cases, except for LSF Vector IX the reference structure is obtained closely and compliance values change very little or not significantly compared to LSF Vector II. However, the number of evaluations is reduced for LSF Vectors III, IV, VII and VIII. In these cases, adding physical variables such as strains and stresses contributes positively by reducing the effort of the learning task, as they represent processed displacements that are directly related to the strain energy density of the element.

LSF Vector V, VI, show that coordinates and averaged displacements do not reduce the number of evaluations but can even increase the number significantly (LSF Vector VI).

When only coordinates are used as model inputs (LSF Vector IX), it is possible to obtain a useful structure, however the boundaries are blurred. In this case, the model learns a separate update-signal value for each element of the structure independent of the state. This results in the highest computational cost and objective value of this study.

For the SVR-TOPS, the results indicate that adding meaningful LSFs can reduce the computational cost for learning. Accordingly physical variables about the state of the element can serve as useful additional information. Otherwise, assuming the LSFs ultimately represent the same information, longer LSF vectors will not necessarily lead to an improvement of the objective or to a reduction of learning effort, but can actually demand more samples for learning.

5.3.6 Experiments with Aggregated Sampling

The idea behind the aggregated sampling approach is proposed in Sec. 4.4.2. It enables the optimization of noisy objective functions that render finite differentiation inaccurate. For empirical evaluation, SVR-TOPAS is run for different group sizes $N_{\mathcal{G}}$ in order to empirically study the effect of this parameter on the optimization. At first, it is applied to the reference problem as in Sec. 5.1. The more challenging learning task with LSF Vector II is considered.

Secondly, the experiments explore the effects of imposing noise on the

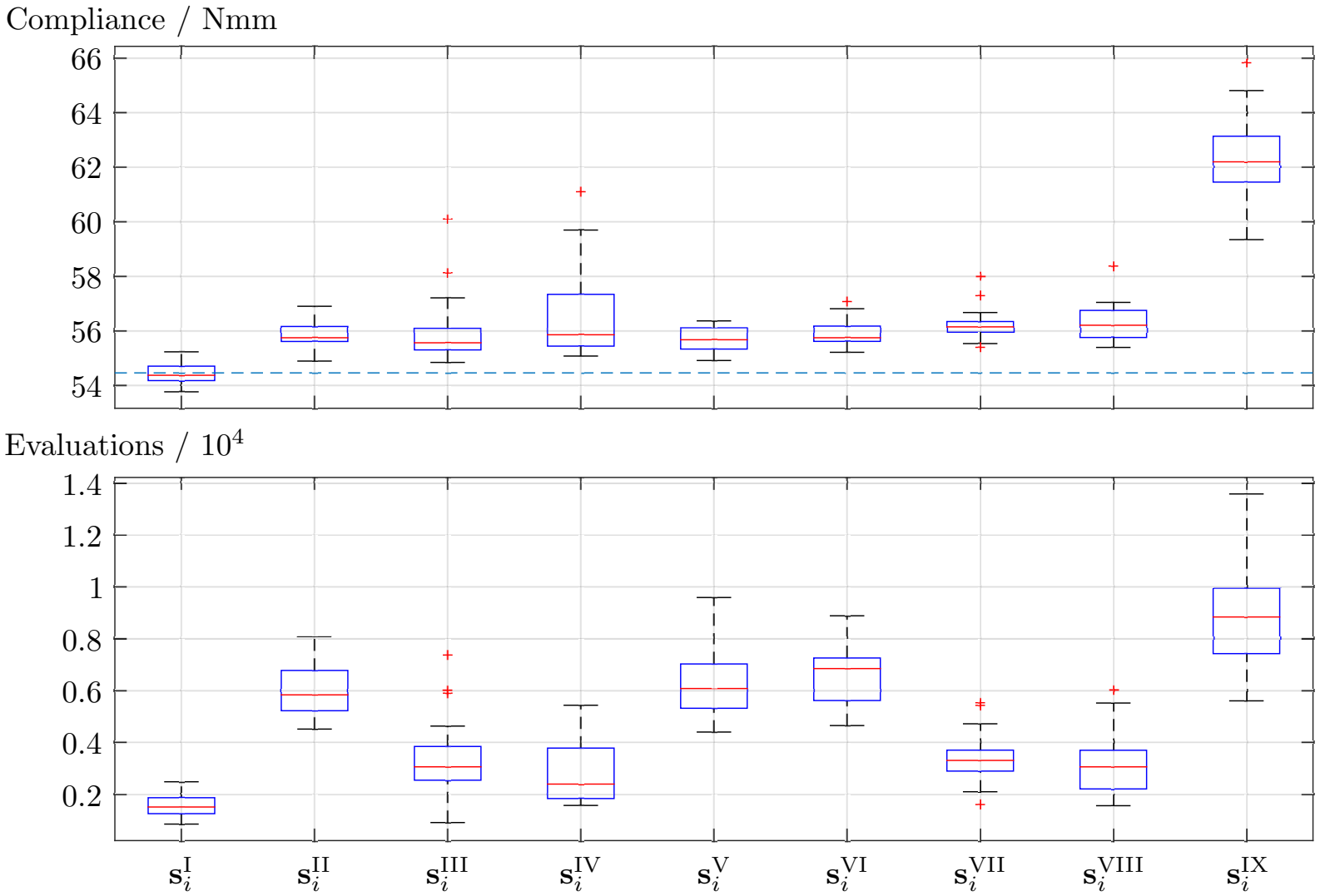
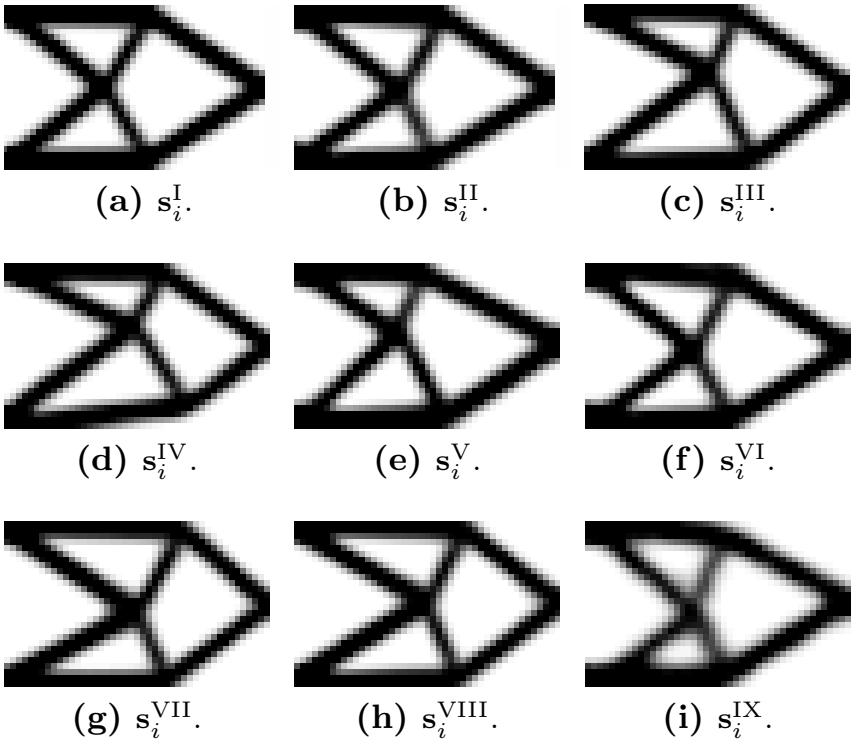


Figure 5.21: Resulting compliance (top) and required evaluations (bottom) of SVR-TOPS with LSF Vectors I to IX. The dashed line shows the compliance of the reference structure.



(a) $\mathbf{s}_i^{\mathrm{I}}$. **(b)** $\mathbf{s}_i^{\mathrm{II}}$. **(c)** $\mathbf{s}_i^{\mathrm{III}}$.

(d) $\mathbf{s}_i^{\mathrm{IV}}$. **(e)** $\mathbf{s}_i^{\mathrm{V}}$. **(f)** $\mathbf{s}_i^{\mathrm{VI}}$.

(g) $\mathbf{s}_i^{\mathrm{VII}}$. **(h)** $\mathbf{s}_i^{\mathrm{VIII}}$. **(i)** $\mathbf{s}_i^{\mathrm{IX}}$.

Figure 5.22: Optimized structures of SVR-TOPS with alternative LSF vectors, including vectors I and II for completeness.

objective function. In this scenario the compliance objective function $c(\boldsymbol{\rho})$ is replaced by an objective function $\tilde{c}(\boldsymbol{\rho})$ which is subject to numerical noise:

$$\tilde{c}(\boldsymbol{\rho}) = c(\boldsymbol{\rho}) + z, \text{ with } z \sim \mathcal{N}(0, \sigma_F^2) \ ,$$

with a normally distributed random variable z with zero mean and variance σ_F^2 added to the compliance. The experiment is run for $N_{\mathcal{G}} = 1, 2, 5, 10, 20, 30, 40, 50, 75, 100$ and noise is added with $\sigma_F^2 = 0, 0.0025, 0.005, 0.01, 0.02, 0.03$. The resulting structures and the relation between noise and optimized compliance for different group sizes is shown in Fig. 5.23 and Fig. 5.24, respectively.

The case of no noise and $N_{\mathcal{G}} = 1$ corresponds to SVR-TOPS and exactly the same result as in 5.3.2. is obtained. Generally, it can be seen that with an increasing value of $N_{\mathcal{G}}$ the average optimized objective value is increasing. This trend of increasing compliance is expected, since the data and consequently the sensitivity model become less precise. However, even for large group sizes such as $N_{\mathcal{G}} = 50$ (corresponding to about 4% of the elements) still a reasonable structure is obtained that shows the structural features of the reference.

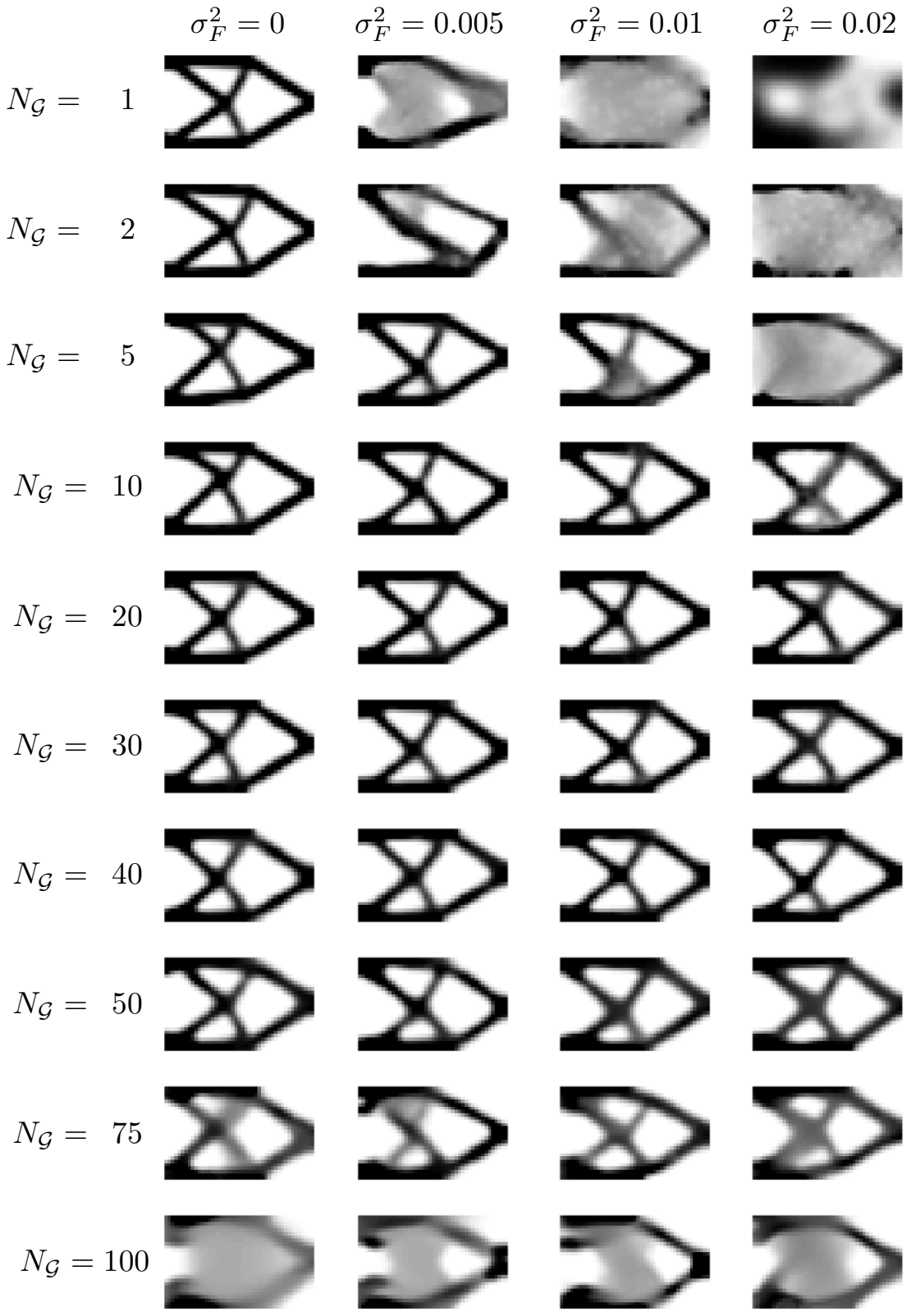


Figure 5.23: Resulting structures from SVR-TOPAS for increasing values of the group size parameter $N_\mathcal{G}$ (from top to bottom) and increasing noise (from left to right). The aggregated sampling enables to obtain the reference even when the objective function in subject to noise.

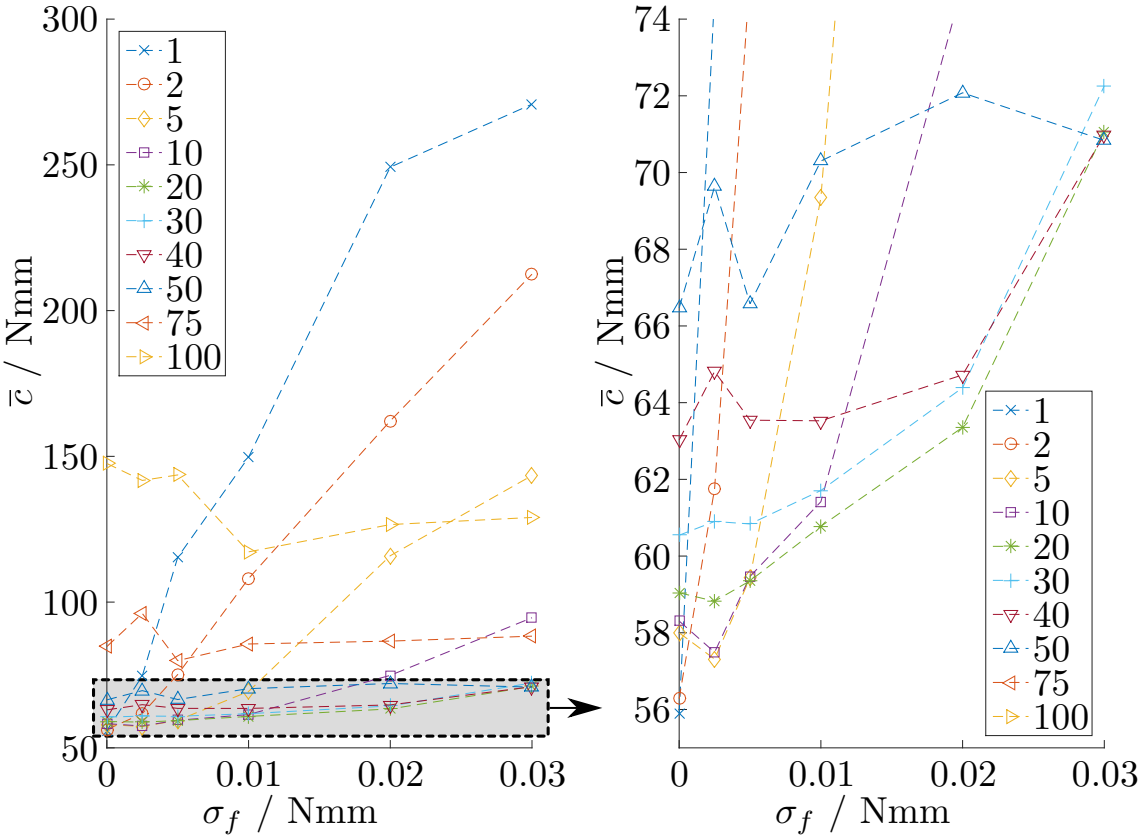


Figure 5.24: The variance of the noise versus the mean compliance of the optimized structures obtained from running SVR-TOPAS for several group sizes $N_\mathcal{G}$. The right plot is zoomed in on the y-axis. For low variance a low group size yields best results, as noise is increasing, higher group sizes are superior.

It is notable that the SVR-TOPS approach without aggregation reacts very sensitive to the noise and even for the lowest tested noise level no converged result is obtained. For each $\sigma_F^2 > 0$, there is a value of $N_\mathcal{G}$ that achieves a trade-off between the errors introduced by the noise and groups sampling. Table 5.4 shows the group size that results in the lowest mean compliance for the given noise variance. For relatively high levels of noise the optimization fails to find a meaningful structure for small values of $N_\mathcal{G}$.

The results are in accordance with the theoretic discussion in Sec. 4.4.2. In the tested example, the influence of the error introduced by averaging over elements, which are close in LSF space, is lower than the benefit from reducing the noise. This demonstrates empirically that the aggregated sensitivity with $N_\mathcal{G} > 1$ can achieve a more robust optimization against normally distributed random noise in the objective function.

5.4 Overall Discussion

The proposed generic optimization techniques are able to reproduce the reference structure, however they vary in the accuracy and computational cost associated with learning and as such have different potential use cases. An important result is that the reference structure can still be reproduced even from LSF Vector II, i.e. when the strain energy density is excluded. Table 5.5 provides an overview of the different modelling approaches and the main findings.

For LSF Vector I, NE-TOPS requires notably fewer evaluations than PCM-TOPS when the model complexity is chosen adequately. PCM-TOPS has difficulties when the prototypes are only distributed sparsely in LSF space. For LSF Vector II, interestingly, NE-TOPS achieves higher model correlation to the sensitivity than PCM-TOPS, but in contrast to PCM-TOPS the designs do not reproduce the reference. Thus, in the case of LSF Vector II, the ability to represent the structure for evolutionary optimization seems to be more relevant than an accurate sensitivity prediction.

Albeit the CMA-ES is an efficient optimizer, the number of evaluations for the evolutionary learning approaches is high compared to the supervised approach. For LSF Vector II, the update-signal model is rarely reused and, furthermore, many model parameters are required to obtain a suitable model at all. On the reference example with moderate mesh size, the computational cost can even be higher than for a naive finite differentiation approach (i.e. higher than M_{ref}), although this would change for larger mesh sizes. The parameters M and $M^{\max}$ imply lower limits on the required numbers of evaluations and provide a possibility of tuning the algorithm for higher efficiency. For the sake of comparability of the experiments in Sec. 5.2, these parameters were chosen identical for LSF Vectors I and II. For the relative simple learning task for NE-TOPS with LSF Vector I and a network model with few parameters, it would be possible to notably reduce the number of evaluations by reducing M and $M^{\max}$.

Table 5.4: Group size parameter for SVR-TOPAS that results in the lowest compliance for a given noise value.

Noise variance	σ_F^2	0.0025	0.005	0.01	0.02	0.03
Best group size	$N_{\mathcal{G}}$	5	20	20	20	50

Table 5.5: Overview of TOPS approaches and findings.

Name	Update-Signal Model	Findings
NE-TOPS	Multi-Layer Perceptron, tuning of network weights and biases by CMA-ES.	Efficient when one of the LSF is linearly related to sensitivity. Good correlation of update-signals with sensitivity. Computationally most expensive method for LSF Vector II. Best objective values are obtained by reducing intermediate densities.
PCM-TOPS	Piecewise-constant model, tuning of update-signals for clusters of elements by CMA-ES.	Best reproduction of reference structure for LSF Vector II without usage of FD samples. Most effective for small LSF spaces, less when prototypes are sparse in LSF space. Can be interpreted as a representation of groups of elements with similar state.
LIN-TOPS	Regularized linear regression, solving least squares problem based on FD samples dataset.	Cheapest and very effective method for cases with linear relation between LSF and sensitivity. Non-linearities can only be handled by feature construction.
SVR-TOPS	ϵ-SVR with RBF kernel, solving quadratic optimization problem based on FD-samples dataset.	Cheapest approach for LSF Vector II, can handle nonlinear relations of LSF to sensitivity well, hence robust to LSF choice. Mesh-independency of evaluations and design demonstrated.
SVR-TOPAS	Same as for SVR-TOPS, but FDs are determined using aggregated sampling.	Similar to SVR-TOPS for LSF Vector II, but can handle moderate amounts of noise in the objective function.

However, for cases that require a high model complexity, a reduction of M and $M^{\max}$ would lead to deterioration, as it reduces the chances of the CMA-ES to find adequate model parameters.

LIN-TOPS can be very efficient for LSF choice, but depends on the construction of adequate additional LSFs. Most reliable is SVR-TOPS: it is robust to the choice of the LSFs and it is able to reproduce the reference structure accurately. Deficiencies in the LSFs are simply compensated by additional sensitivity samples. Even for the more difficult test case, it is being drastically more efficient than a naive finite differentiation approach, a fact that is especially important when the mesh size increases.

The supervised learning approach yields more accurate results due to the explicit sampling of the sensitivity function. Additionally, it requires a lower number of evaluations and moderate levels of numerical noise can be tackled by aggregated sampling. Thus, when sufficiently precise sensitivity estimates by finite differences are possible, the supervised approach should be preferred. The situation is different in case of objective functions that are non-smooth or involve severe multi-modality: When no precise finite difference samples can be obtained or a gradient descent is unsuccessful, the evolutionary learning is a more robust and possibly the only working option.

The results in this chapter are based on a single compliance reference test case and leave a necessity to briefly comment on possible generalization of the results when changing the objective function. Within the domain of compliance minimization, the underlying sensitivity function (5.1) is not changing with the boundary conditions, i.e. loads and support. Hence, in this cases the learning task of the model remains unchanged. Therefore, qualitatively similar results in terms of compliance, evaluations and visual reproduction of baseline structures can be expected when changing the load case and supports. This also has been observed in experiments not documented here, but for instance some alternative results with a different configuration can be seen in [14] (although the algorithm has been set-up slightly differently). Certainly, future work should validate this by additional experiments. When changing the objective function, and especially when consideration of more challenging, non-linear problems, the transfer of conclusions is maybe possible, but certainly more risky, implying an even bigger need for more experiments in future research. Nevertheless, Table 5.5 provides rather abstract conclusions that are expected to keep some meaning for general use cases, at least to choose the most promising approach. The next chapter applies PCM-TOPS in the domain of crashworthiness, supporting the general applicability of the approach on a

fundamental level.

A potential idea when considering a model is to reuse it in other problem instances that target the same objective function, but for instance different load cases. Intuitively, this should be possible when the formulation of the analytical sensitivity is unchanged. However, at least the presented experiments indicate that this hypothesis has to be rejected. In Fig. 5.5 and 5.16 it can be seen that even for the simple case of strain energy density as a LSF, the model does require re-learning several times during the optimization. Certainly, if it is assumed that the learning task consists in general of identifying a LSF as sensitivity information, this can be done in a simple trial and error set-up, and would possibly provide an interesting LSF selection method in context of the presented approach. However, in more general cases, such as LSF Vector II, the learning of a non-linear relation is expected and the retraining becomes even more important as can be seen in Fig. 5.10 and 5.18. Here, it can be seen, that a model that performs well in more than local region of LSF space is not obtained. Then, the target of a more global model is a serious challenge that probably requires a more complex model and even higher learning effort.

Generally, an approach may fail, deteriorate or at least rise noticeable in computational cost when the relation of LSF to the sensitivity cannot be modelled well, such as in the case of the LIN-TOPS or NE-TOPS with LSF Vector II. Partially, this can be compensated by frequent retraining, but the preferred solution is to gather additional LSF that contain missing or more explicit sensitivity information. This may be preferably done by a basic or improved analytic modelling of the problem to identify (adjoint) state information that can be used as LSF. It is likely that these *features*, if once shown to be useful for a certain objective function, could be generalized to different boundary conditions.

5.5 Summary of Contribution

This chapter presents evaluations of the proposed generic topology optimization approach: Topology Optimization by Predicting Sensitivities (TOPS). We motivate a minimum compliance reference problem and chose exemplary Local State Feature vectors. Evolutionary and supervised approaches for learning the update-signal models are considered. Statistical evaluations for all approaches provide a foundation for comparisons and discussions.

All TOPS variants are able to reproduce the reference design without

pre-defined gradient. By comparing different feature vectors, it is shown that this is even possible without explicit knowledge about the strain energy density. These findings highlight the feasibility of the learning component as part of the topology optimization.

Evolutionary learning of a piecewise-constant model is more capable of efficiently modelling the required non-linear relations, compared to evolutionary learning of the artificial neural network. The latter is suitable to easily chose from several input features. The supervised learning approaches, especially support vector regression, show superior characteristics in terms of reliability, accuracy and computational cost compared to evolutionary learning. The number of evaluations required by the method is drastically lower than for naive finite differentiation, especially when the mesh size is increasing. This is obtained under the assumption that accurate finite-difference samples are possible. However, the method can also handle small amounts of numerical noise by an aggregated sampling approach.

We conclude that TOPS with supervised learning is the preferred method for single-modal and smooth objective functions. TOPS with evolutionary learning can be applied when sensitivity samples are imprecise or gradient-descent is failing. With the target of crashworthiness topology optimization, we apply it to such a problem in the next chapter.

6 Topology Optimization of Crashworthiness Objectives

Within the automotive industry, the task of designing optimal components for crash scenarios is of high practical relevance in the vehicle design process. Therefore, and due to the difficulties it poses to existing topology optimization approaches, it is chosen as a first domain of application of the proposed generic approach.

This chapter is structured as follows: Section 6.1 briefly describes the approaches for crashworthiness topology optimization from the literature on a conceptual level. Section 6.2 presents the application of PCM-TOPS to a problem of energy absorption maximization. The considered example is a clamped beam subject to a transversal bending crash load. Subsequently, the same algorithm is addressed to optimize the thickness distribution of a 3-dimensional thin-walled frame structure in Sec. 6.3. The objective function is changed to the target of minimizing intrusion to demonstrate the adaptivity of the learning concept. Comparisons to a state-of-the-art uniform energy heuristic are presented. Parts of this chapter are based on [10, 13].

6.1 State-of-the-art

Besides many requirements such as driving dynamics, structural statics and dynamics, NVH (noise vibration harshness) or acoustics, passive crashworthiness safety is an important part of the virtual multidisciplinary vehicle design process in today's automotive industry [57]. Crash requirements are defined by programs such as Euro NCAP[1] in Europe, US-NCAP in the US and comparable programs around the world. Typically, the fulfilment of these requirements is measured by a safety ranking as an important indicator for vehicle safety and thus automakers aim at designing new vehicles to perform well in the defined tests.

[1] NCAP: New Car Assessment Program

Due to the expense of real prototype testing especially in crashworthiness scenarios, numerical simulations provide a possibility for significant cost reduction. However, modelling comes with challenges as it has to take into account physical non-linearities, for instance large deformations, material subject to non-elastic strain, up to failure, or multiple contacts. This results in complex and computationally expensive simulation models. Yet, with increasing simulation capabilities, crashworthiness simulations have become an important step of the virtual vehicle design process. In parallel, there is significant pressure to shorten product cycles and the necessity of reducing fuel consumption. This increases the desire to find the most efficient lightweight vehicle components for crashworthiness. Thus, the interest in optimization is increasing.

In *sizing* or *shape* design optimization problems a limited number of optimization parameters can be addressed with global search algorithms [57]. However, the field of crashworthiness *topology* optimization is still relatively young and an active area of research. Challenges include the high dimensionality inherent to topology optimization and the complexity of crash simulations that involve model non-linearities, non-smooth structural responses or even bifurcations. As a result, objective functions that formalize engineering targets usually contain numerical noise and considerable non-linearities. Typical targets in crashworthiness optimization of vehicles are for instance high energy absorbing components, acceleration measures such as the head-injury criterion, smoothness of the force-displacement curve and low/high stiffness of special parts, depending on the vehicle component and crash scenario that are considered [157].

Existing gradient-based methods are only applicable for considerably simplified problems and as a consequence, heuristic approaches for crashworthiness topology optimization have been developed. These will be discussed on the next pages showing that these rely on arguable assumptions and are only applicable to specific use cases and design criteria.

Early research on topology optimization for crashworthiness was conducted in the late nineties by Mayer et al. for maximizing crash energy absorption based on a homogenization approach of a piecewise linear elastic-plastic material model [117]. In their work, optimality criteria are derived based on an objective function that weights the strain energies at specified times.

This idea of a simplified problem with a rigorous mathematical model is reflected in the ground structure approach by Pedersen [134–137]. In the

ground structure approach, the design space consists of a frame/ground structure of beam elements. A simplified crash behaviour is modelled, for instance involving large and plastic deformations. Gradient-based optimization can be performed based on rigorously derived sensitivities. The thickness and/or presence of the beams is optimized for compliance subject to path dependent response [136], desired energy absorption history [134] or desired acceleration history [137]. However, the method is limited to significantly simplified formulations that facilitate the derivation of analytic sensitivities.

A heuristic alternative is the graph based topology optimization approach. It originates from the idea of combining shape optimization with topological changes as presented in the bubble method by Eschenauer et al. [60]. In the bubble method, the position of a small hole or "bubble" is determined by a characteristic function. The bubble introduces a topological change and a consequent shape optimization of the bubble and other variable boundaries is performed. This is done in a loop until a maximum number of bubbles is introduced. The bubble method has been applied to crashworthiness structures using analytic expressions for hole positioning considering maximization or minimization of stiffness [157]. It also inspired topological derivative methods that are used for instance for hole nucleation or reintroduction of solid material in the level set methods [3, 125].

The idea of a graph-based topology optimization is one of the possible representations in the field of Evolutionary Computation as introduced in Sec. 3.3. Independently of EC, a graph-based approach for crashworthiness topology optimization was proposed by Olschinka and Schumacher [126] and considerably advanced by Ortmann and Schumacher [128] for extrusion structures. Similar to the bubble method, topological changes and size optimizations are performed iteratively. However, in the existing graph-based methods applied to crashworthiness, the topological changes are based on heuristic rules from engineering crash experts: for instance "Support Fast Deforming Walls", or "Balance Energy Density" [128]. After each modification of a topology a subsequent shape optimization is performed. Since the shape optimization uses global search, it can account for the actual objective. Yet, the heuristic rules for topological changes are specifically tailored to the problem. By the application of several rules in parallel and by selection of the best performing one, a step into the direction of evolutionary optimization of the topology has been done [127].

Another approach is crashworthiness topology optimization using

Equivalent Static Loads (ESL). ESL methods replace crash loads by one or several computationally cheaper static load cases. The models subject to linear static loads can be addressed with standard gradient-based topology optimization methods. The approaches can be differentiated mostly by how the ESLs are found. Static loads can simply be applied in the regions where the impacts occur [46]. Cavazzuti et al. [38] linearised a frontal crash by substituting inertial forces by equivalent static loads for automotive chassis design. Volz devised ESLs based on physical models considering stroke-energy relation and consequently forces occurring in the crash [195]. The ESLs are designed for different stages that reflect the dynamic nature of the crash. It is applied to topology optimization in the early concept phase of car body design. ESLs can also be determined by multiplication of the stiffness matrix of a linear model with the displacements occurring in the non-linear crash analysis for each of the time steps [98, 130]. The static load cases are combined in a multi-load case topology optimization. Repeatedly, the optimized structure is analyzed utilizing the non-linear analysis and a new set of static loads is obtained. This process is iterated until convergence. It is expected that good results can be obtained mostly for moderately non-linear problems. Some first applications for topology optimization are given in [206, 207, 216]. It is not clear in ESL methods, how much the optimization of the ESLs corresponds to the optimization of the actual objective function defined by the non-linear, dynamic load.

Tovar proposed a Hybrid Cellular Automata (HCA) algorithm as a model for the remodelling process in human bones [190] (compare also Sec. 4.1.3). A cellular automaton consists of cells arranged on a regular grid. Iteratively, cells are updated based on the state of neighbouring cells. In the HCA approach, the states of the cell are energy density and material density. A control-based rule is applied that targets uniform distribution of energies throughout the structure and updates the material distribution accordingly. Although it does not optimize an objective function in the mathematical sense, HCA can be utilized as a method for topology optimization. It leads to useful structures from an engineering perspective for instance minimizing compliance [190, 191]. The working principle assumes that an optimum structure distributes the load uniformly throughout the structure. This is similar to the concept of fully stressed designs [133].

A further development by Patel [131, 132] demonstrated the usefulness of HCA when applied to engineering design optimization of crashworthiness structures. The assumption of the uniform distribution of internal energy densities from elastic and plastic strains can result in structures

with preferable crash characteristics. HCA has also been applied to cost and mass (topometry) optimization of sheet metal structures [121, 122]. A mathematically quite similar topology optimization approach for crashworthiness has also been developed in parallel by Forsberg et al. [66].

Huang et al. proposed the BESO (or SERA) approach to maximize energy absorption of a structure [90, 91]. It is based on internal energies as well. For BESO, simplified analysis is performed to obtain a "Sensitivity Number" that is applied in the optimization of a structure for maximization of energy absorption. The sensitivity number is determined as the internal energy density and elements are removed from the design space that is initially completely filled with material (see the introduction of BESO in Sec. 2.1.3).

In recent developments, a HCA compliant mechanism approach for controlled energy absorption behaviour is applied to optimization of progressively folding thin-walled structures [24, 108, 165]. Since the uniformity assumption is similar for compliance and crashworthiness cases, it can be used for concurrent multi-disciplinary optimization of crashworthiness and stiffness load cases. This was demonstrated by case studies of the author by means of a Scaled Energy Weighting HCA approach [16–18]. HCA has been based on regular cubical design space discretizations. However, realistic behaviour of vehicle components is often much better represented by thin-walled structures. In crash events, thin-walled structures show a significant amount of buckling, and form local, plastic hinges. As consequence, energy is in fact not uniformly distributed. To account for this, a HCA based on macro cells that each includes a set of finite elements has been proposed [58, 92]. Energy can be distributed uniformly over the macro cells, while it can be non-uniform within a single macro cell. Generally, the HCA is limited by the assumption that the uniformity of the field variable leads to an optimal design.

The field of crashworthiness topology optimization is still young and more research is needed to validate existing and new methods to ultimately determine suitable methods for different industrial problems. Especially, comprehensive and conclusive comparisons are still missing. Currently existing heuristics are based on problem simplifications and/or assumptions on the optimum solution and do not provide flexibility on the choice of the objective function. However, this underlines the necessity of studying generic approaches, which enable a clear and conscious choice of the objective function. In the following two sections, TOPS is applied to two crashworthiness topology optimization problems. To facilitate a classification of results we compare to a baseline obtained by HCA. HCA is efficient

and available as commercial tool that is already in use in the automotive industry. Also, HCA is conceptually similar to TOPS in the sense that it is a density-based topology optimization approach as well, however *without* the ability to adapt to the objective function.

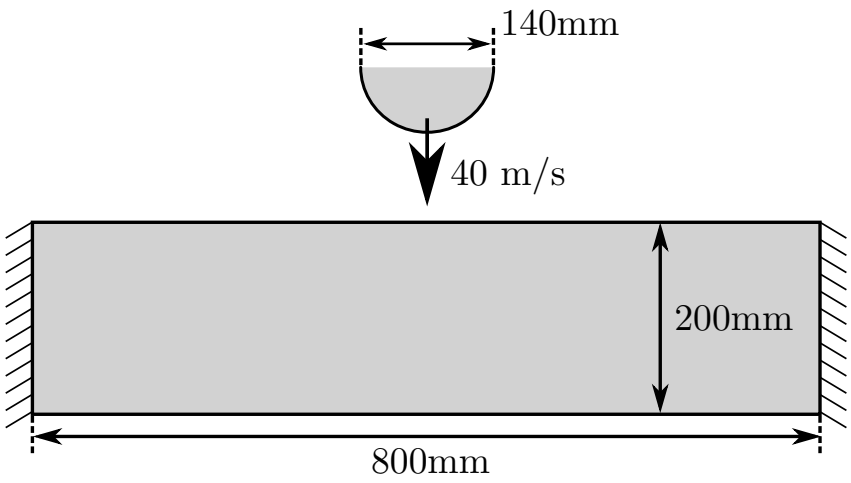
6.2 Maximum Energy Absorption Beam

6.2.1 Beam Model and Optimization Set-up

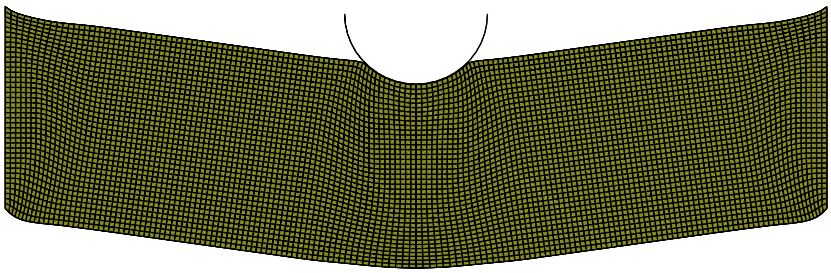
A typical target from crashworthiness topology optimization is the maximization of energy absorption, for instance, in preventing transfer of energy to passengers or sensitive vehicle components in a crash. Also, in bagatelle crashs, directing energy absorption towards specialized components potentially enables an economically efficient replacement in order to reduce insurance cost.

The simulation of a rectangular aluminium beam subject to transversal bending crash caused by a cylindrical rigid pole is considered as test case. A schematic drawing is depicted in Fig. 6.1. A simplified 2-dimensional problem is considered, hence out of plane displacements are constraint to zero. Symmetry of the model is exploited by the definition of appropriate boundary conditions along the central axis, in order to reduce simulation time. One half of the symmetric beam forms the design space and is meshed with 5mm edge length cubic elements, which enable the modelling of the contact. The finite element computations are carried out with the commercial explicit solver LS-DYNA [109]. The LS-DYNA simulation model is of moderate size and can be computed within 15-30 seconds on one CPU of the available computing resources. This enables experimentation and rudimentary statistical evaluation. The specifications of the model are shown in Tab. 6.1. The aluminium material is modelled by a piecewise linear elastic-plastic behaviour with a material interpolation scheme according to [131]. Details of this model are provided in App. C.

For this type of problem, analytical sensitivities are not available, thus standard gradient-based topology optimization methods cannot be applied and it is justified to apply the devised generic TOPS method. Similar problems have been addressed by [35, 91, 131]. Although an academic test case, the dynamic simulation features important characteristics that occur in real-world crashworthiness problems, for instance large and plastic deformations and contacts. Compared to the linear static compliance problem the optimization task increases significantly in complexity and, hence, the



(a) Model.



(b) Deformed state after 2ms simulation.

Figure 6.1: Clamped beam subject to rigid, cylindrical pole crash.

transfer of TOPS to this problem is a major step in the direction of application of the method. We perform a comparison to a reference structure obtained by the HCA algorithm [131] that is available in the commercial implementation LS-TaSC [144].

We define the optimization problem as maximization of the energy absorbed by elastic and plastic deformation of the design space at the final simulation time step:

$$\begin{aligned} \max_{\boldsymbol{\rho}} \mathrm{E}_{\mathrm{abs}}(\boldsymbol{\rho}) &= \sum_{i=1}^{N} \mathrm{IED}_i(t = t_{\mathrm{end}}) v_i \\ \text{s.t. : } & \mathbf{r}(t, \boldsymbol{\rho}) = \mathbf{0}, \\ & V(\boldsymbol{\rho})/V_0 = f, \\ & 0 < \rho_{\min} \leq \rho_i \leq 1, \ i = 1, \ldots, N \ , \end{aligned} \tag{6.1}$$

with the overall energy absorption of the design space $\mathrm{E}_{\mathrm{abs}}(\boldsymbol{\rho})$, the internal energy density of an element $\mathrm{IED}_i(t = t_{\mathrm{end}})$ and the simulation time t. The crash simulation is expressed by the residual error $\mathbf{r}(t, \boldsymbol{\rho})$ of the dynamic finite element analysis.

As for the compliance problem, it is necessary to chose the LSF that are used in the optimization. Due to the time-dependency, possible state information exists at each time step at which the governing equation system is solved. Practically, a subset of the discrete time histories of the LSFs can be sampled, but even then a large number of features can be obtained[2].

Since the objective function is energy absorption, it makes sense to consider the energy absorbed in a finite element as LSF: concretely, the elemental internal energy density IED_i. The internal energy is defined as the energy absorbed under elastic and plastic deformation, i.e. the integral under the stress-strain curve. If the strain is assumed as an independent variable, the internal energy density increases and decreases with the density. Thus, the internal energy density can be motivated as a valuable LSF. To further reduce the number of features, its maximum over all time steps is used, instead of using its value at several specific time steps. Furthermore, the update-signal model should be able to differentiate between low and high density elements, therefore the density is also used as LSF.

[2]Let us consider an element defined by eight nodes with three degrees of freedom each. Then, each element has $8 \times 3 = 24$ displacement LSFs. Sampling the time history every 0.1ms results in 480 basic LSF for the beam simulation.

Table 6.1: Specifications for clamped beam model subject to pole crash.

	Symbol	Value
Pole z-velocity	-	40m/s
Number of elements	N	$80 \times 40 = 3{,}200$
Termination time	t_{end}	2.0ms
Young's modulus	E_0	70,000MPa
Volumetric mass density	ϱ_0'	$2.7 \cdot 10^{-9}\mathrm{t/mm}^3$
Poisson's ratio	ν	0.33
Yield stress	σ_{y0}	241MPa
Tangent modulus	E_{t0}	70MPa
Static/Dynamic friction coefficient	-	0.2

This results in the two-dimensional LSF vector:

$$\mathbf{s}_i = \begin{pmatrix} \rho_i \\ \max_t \mathrm{IED}_i(t) \end{pmatrix} .$$

Due to time and resource constraints, not all TOPS variants could be applied and thoroughly evaluated. Initial experiments with SVR-TOPS and SVR-TOPAS showed only limited potential. Consequently, the level of numerical noise has to be considered higher than what can be handled by the aggregated sampling approach, for the given crash simulation. The effect of numerical noise is especially disturbing when small finite differences are applied to the design. Therefore, sensitivity estimation is unreliable. Furthermore, the significant non-linearities can lead to an early convergence to local optima. These circumstances suggest the choice of TOPS with explicit evolutionary optimization of the model parameters. It has a change to overcome local optima and it is not directly effected by the noise. PCM-TOPS reproduces the compliance reference more reliably and efficient than NE-TOPS, when the relation of the LSF to the sensitivity is non-linear (Sec. 5.2.2). Since we expect substantial non-linearity, PCM-TOPS is chosen as modelling approach. As discussed in Sec. 5.2.3, it works especially well for a low dimensional LSF space. (Compare also the overview in Sec. 5.4.)

Table 6.2: Specifications of PCM-TOPS for beam crash energy absorption.

	Symbol	Value
Evaluations per learning step	M	250
Max. evaluations per iteration	$M^{\max}$	750
Number of prototypes	P	30
Improvement threshold	$\Delta F_{\min}$	0.001
Target volume fraction	f	0.4
Penalization	p,q	3
Filter radius	$r_{\min}$	8mm
Number of offspring	λ	14
Number of parents	μ	7
Minimum density	$\rho_{\min}$	0.05
Move limit	m	0.1
Initial global step size	$\sigma^{(k=0)}$	0.3

Table 6.2 shows the optimization specifications, where the symbols $\rho_{\min}$, f, p were introduced in Sec. 2.2, λ, μ, in Sec. 2.3.1, $r_{\min}$, m in Sec. 4.2, $\Delta F_{\min}$ in Sec. 4.2.1, P in Sec. 4.3.2 and q in App. C respectively, and $\sigma^{(k=0)}$ is the initial step size of the CMA-ES. Important parameters are the numbers for the allowed evaluations M and $M^{\max}$, which were introduced in Sec. 4.2.1. Enabling more evaluations increases the chances of the CMA-ES to find better model parameters. Over the course of the optimization this will lead to more successful updates and, ultimately, to a better result. However, the practical choice of M and $M^{\max}$ is a trade-off that considers simulation time and computational resources. The dimensionality of the evolutionary search is identical to the number of prototypes P. Therefore, on one hand, very large values should be avoided. On the other hand, too few prototypes will limit the ability to represent the structure and reduce the possible design complexity. Thirty prototypes are chosen as a trade-off. The CMA-ES population sizes are set to the default values according to the recommendations for real-world applications [78]. In order to avoid numerical problems elements with $\rho_i = \rho_{\min}$ are removed from the mesh.

For the model optimization, the Python implementation of the CMA-ES by Hansen[3] is used [76]. A LS-DYNA topology optimization interface was developed to couple the optimization algorithm to the commercial solver [13]. It provides functionality such as parsing and writing of input decks, parameterization of material models, and reading of output files to access simulation results, including LSF. The optimization was run for 15 different random seeds. The results are discussed in the next subsection.

6.2.2 Optimization Results

Figure 6.2 shows the objective value versus iteration and the required evaluations. It can be seen that in fact a similar behaviour as for the minimum compliance problem in Sec. 5.2 is obtained. By design of the algorithm, a monotonic increase of the objective function is achieved over iterations. The vertical ascents when plotting the objective versus evaluations show that the update-signal model can be reused several consecutive iterations during the optimization. This is observed for the mean, but especially for the best run, leading to fast convergence in terms of evaluations. On average 6,077 evaluations are required until convergence, yet to reach a state from which the objective hardly changes requires roughly half of these evaluations.

[3] `https://www.lri.fr/~hansen/cmaes_inmatlab.html#python`

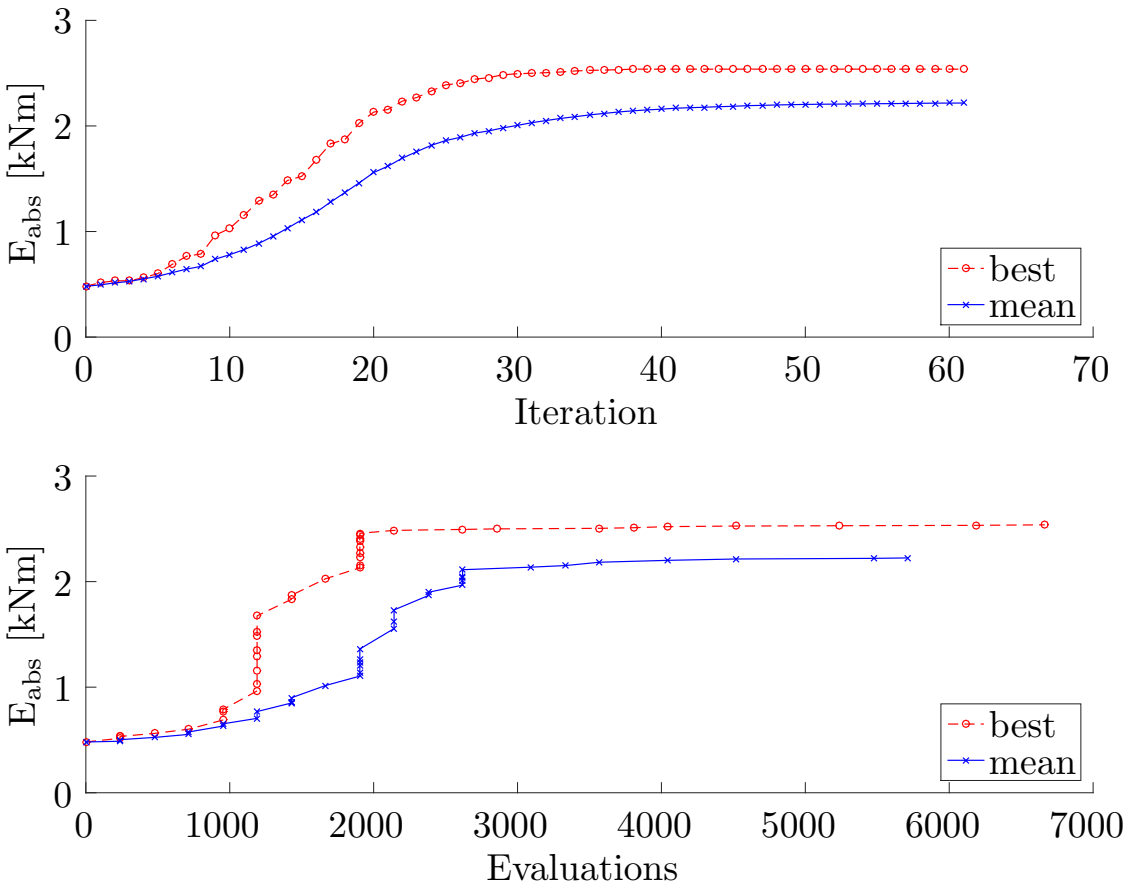
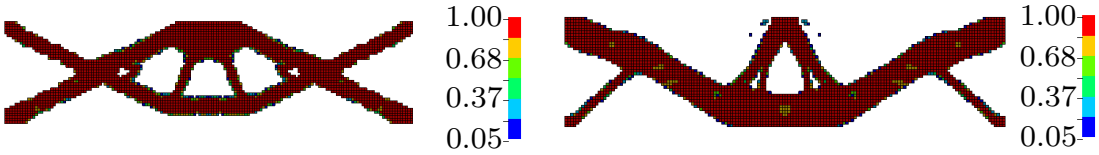


Figure 6.2: Energy absorption versus the number of iterations respectively, evaluations for beam crash energy maximization with TOPS-PCM.

Figure 6.3 shows the resulting optimized structure of the best run in comparison to the baseline obtained from HCA. The results realize different concepts to achieve the energy absorbing behaviour. The best PCM-TOPS structure shows additional beams and also a different thickness distribution.

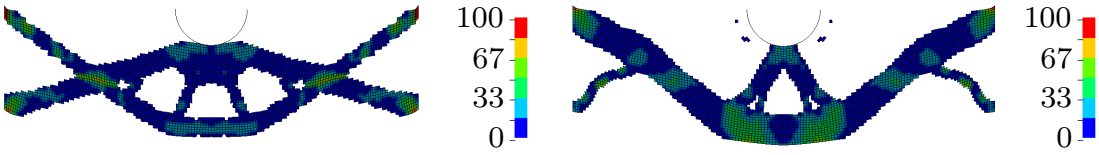
The distribution of internal energies at the final time step for both structures are shown in Fig. 6.4. Both structures show areas of high and low energy densities, yet the HCA structure shows less peaks of very high energy densities, which can be expected as its underlying target is uniformness of energy densities.

According to (6.1), we quantify the performance by energy absorption. In this experiment, the HCA result, which is obtained after 100 evaluations achieves a higher energy absorption of 2.93kNm. The best structure of TOPS-PCM achieves 2.54kNm and on average 2.21kNm. This confirms that HCA is an adequate heuristic for the considered problem, when en-



(a) PCM-TOPS. **(b)** HCA.

Figure 6.3: Optimized structure for beam crash energy maximization, the density is colour-coded.



(a) PCM-TOPS **(b)** HCA.

Figure 6.4: Optimized structure for beam crash energy maximization, the internal energy density of the final time step is colour-coded in Nmm/mm^3.

ergy absorption is the target of the optimization. Yet, the PCM-TOPS results are obtained without the previous knowledge on the benefit of energy uniformness, a fact that is compensated by computational efforts. Another potentially beneficial aspect of PCM-TOPS is that several alternative structures with varying degrees of complexity are obtained shown in Fig. 6.5 that provide different energy absorption concepts that can be studied by experts. The large number of solutions with high energy absorption are most likely caused by multi-modality of the problem, underlining an effect of the various problem non-linearities.

To evaluate the relation of the LSFs and the update-signals, the empirical correlation coefficient is shown in Fig. 6.6 (More information and the coefficient's definition are given in App. A.4). For the density LSF, there is a trend of increasing correlation. This is intuitive, as in the early iterations the elements have similar densities that cannot have a large effect on the update-signal. As the structure converges, elements with a higher density are more often increased or kept, while the opposite happens for elements with low densities. A high internal energy density LSF favours an increase of the elemental densities yet the update-signals are far from modelling a linear relation. Plots of the actual update-signal model are shown in App. C in Fig. C.2.

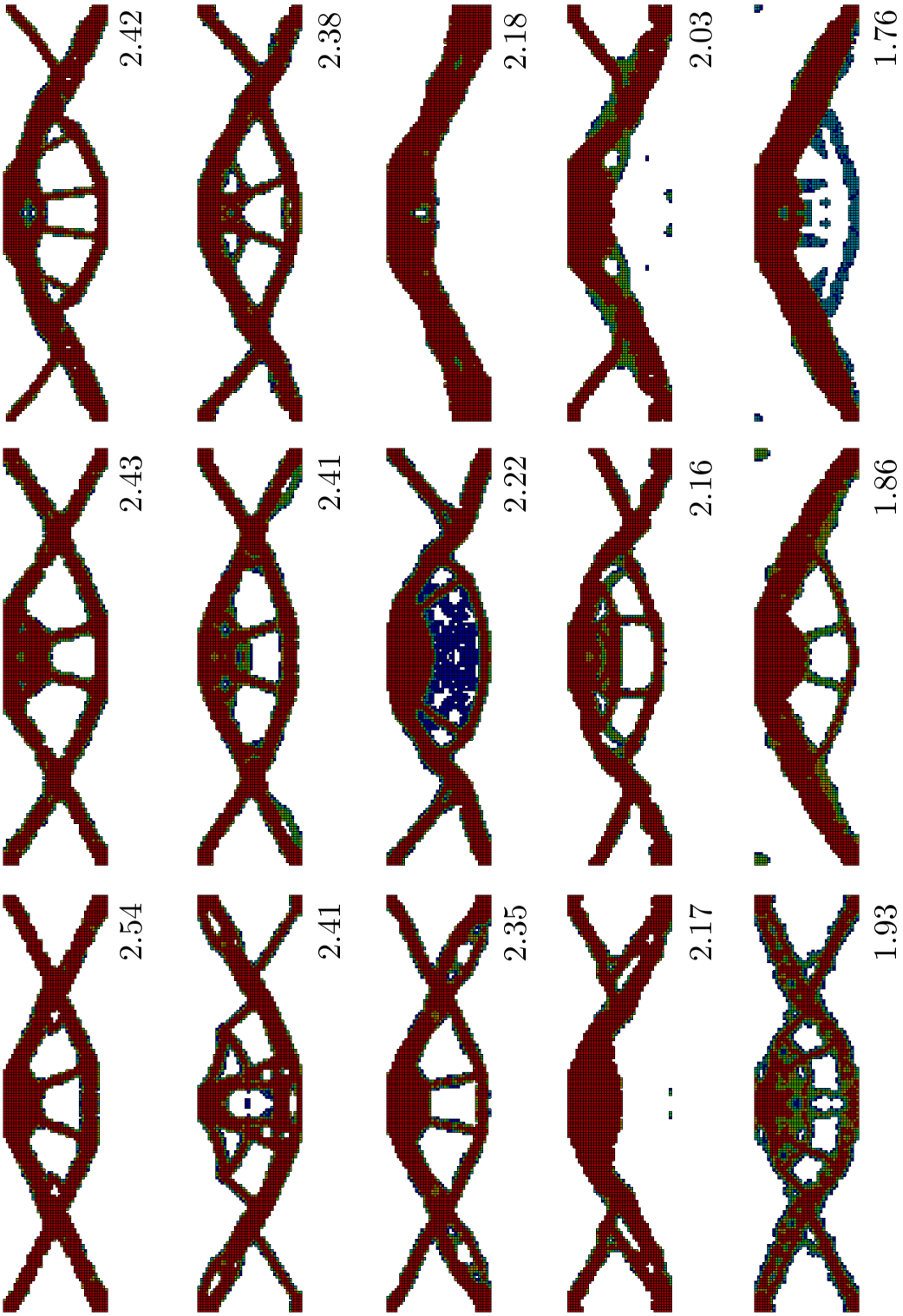


Figure 6.5: Optimized structures obtained by all 15 PCM-TOPS runs for the maximum energy beam, ordered by decreasing E_{abs}/kNm, starting from the highest.

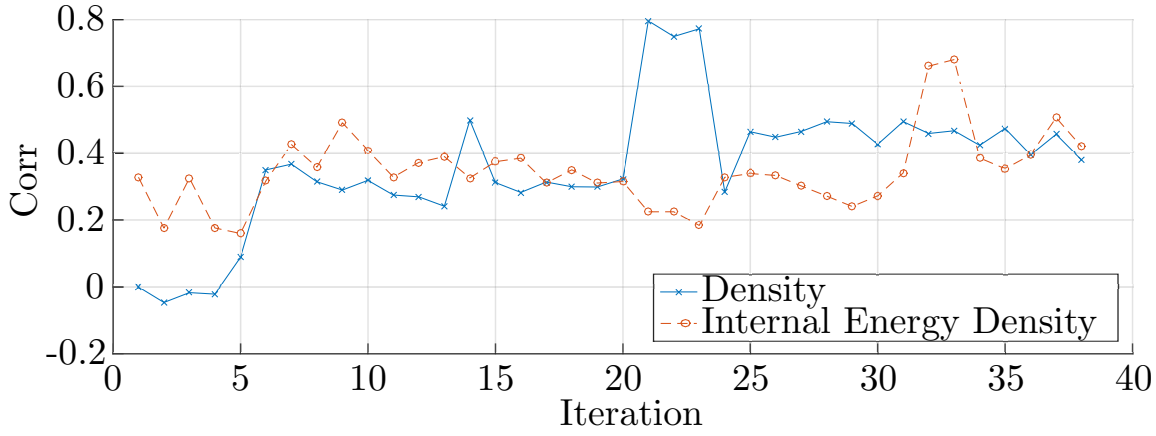


Figure 6.6: The empirical correlation coefficient between update-signals and LSFs plotted over the iterations for the beam structure optimization. The LSFs are weakly to moderately and sometimes highly correlated to the update-signals. Since the optimization problem is formulated as maximization, a high positive correlation coefficient implies that for a large LSF value the density is rather increased than decreased.

We further study the best run to illustrate the working principle of the method. Fig. 6.7 shows the colour-coded update-signals mapped onto the beam structure for different iterations, where positive values are cut, since they cannot be handled by the OC-update. It shows regions in blue where the density is to be reduced, and elements in red where the density should be increased. Intermediate values can result in both depending on the distribution of update-signal values over the complete design space. Elements with the same colour form one cluster in LSF space, although some colour values are closely nearby and cannot be distinguished visually. Interesting topological patterns can be observed, especially in the first iteration. The update-signal model optimization quickly identifies rough areas where material is removed, which around iteration 13 provides a basis structure that is then slowly refined. Elements belonging to one cluster are often distributed over the complete structure, some clusters appear to be distributed noisily over the structure. The sometimes strong pixelization of the clusters towards the end of the optimization indicates that a smoothing of the LSFs over iterations might be useful. Smoothing methods are already in use for example in the stabilization method in BESO [90] or the element memory in HCA [131].

Overall, the results show that the proposed generic TOPS method can be transferred from the minimum compliance problem directly to a problem of crashworthiness. In this case, HCA provides a higher energy absorption

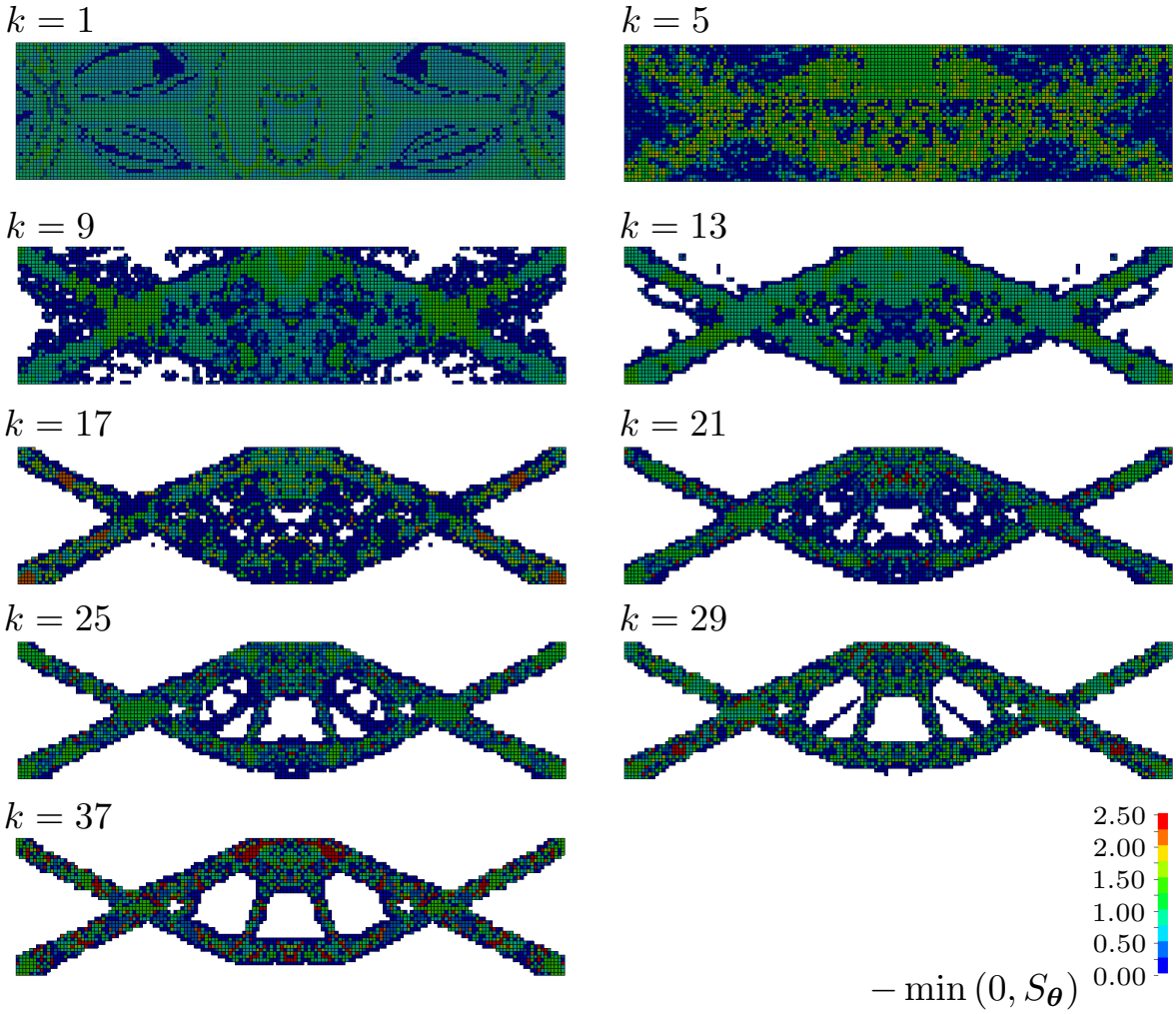


Figure 6.7: The optimized update-signals colour-coded onto the beam structure for different iterations, where blue is a signal for reduction and red for increase of the density, respectively.

than TOPS and the result confirms the benefit of the energy uniformness assumption at least for the given objective function. However, TOPS performs qualitatively similar and provides meaningful structures *without* the previous knowledge of targeting energy uniformness throughout the structure.

6.3 Minimum Intrusion Frame

6.3.1 Frame Model and Optimization Set-up

In this section, we consider a simulation model with higher relevance in practice that is inspired by a side pole Euro NCAP crash test [61]. The side pole impact is a dangerous crash scenario, in which the vehicle is moving laterally into a pole-like structure with high risk of injuries of the passengers due to intrusion of the pole and vehicle body components towards the cabin. Structural components that are involved in this scenario are the side sill, the door and the a/b-pillars. For a favourable passive safety the life cell within the cabin should be intruded as least as possible. Figure 6.8 illustrates the problem of intrusion in the side pole crash scenario.

In the side pole impact as well as in many other crash scenarios, it is desired to absorb energy within special components or areas of the vehicle and protect other components or areas from deformation. From the perspective of optimization, in these scenarios, minimal or at least constraint displacements are desired in order to achieve stiff behaviour at selected locations. This implies a change of the objective function from maximizing an energy absorption measure to minimizing a measure of intrusion.

We consider a simulation model of a 3-dimensional aluminium frame

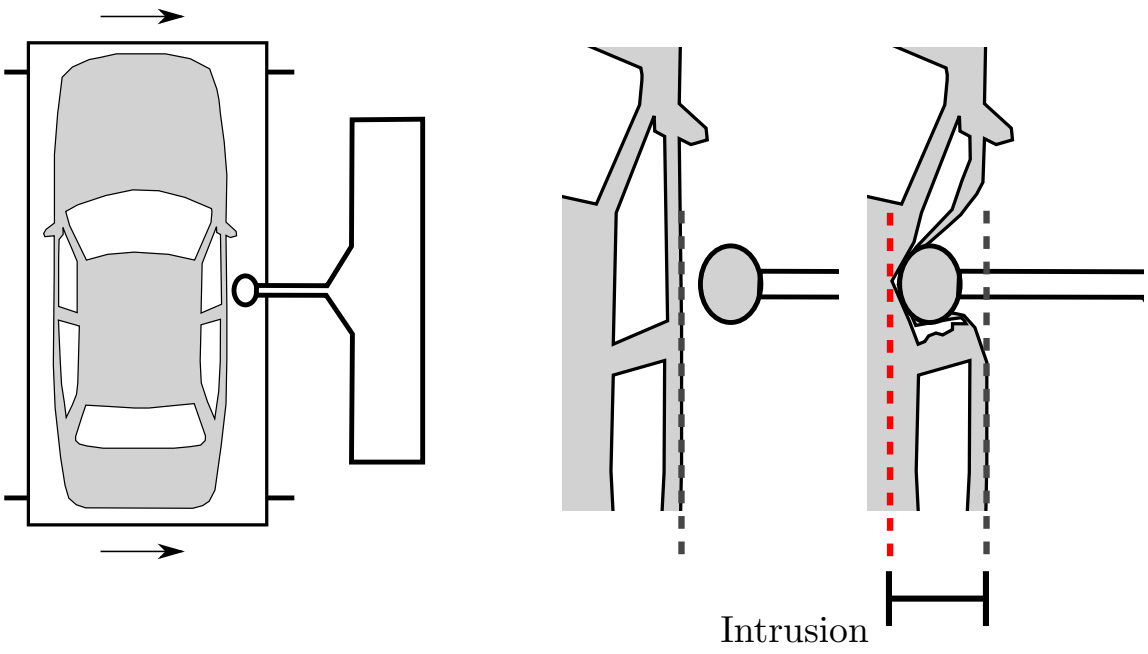


Figure 6.8: Illustration of intrusion during a side pole impact.

supported at both ends that is crashed with a rigid cylindrical pole, as shown in Fig. 6.9. The whole frame is the design space and is discretized with plane stress shell finite elements with edge length of 4mm. The thin-walled model is able to capture the buckling behaviour of typical vehicle components that undergo large deformations. In this case, the elemental density design variables refer to elemental thickness. A material interpolation scheme as in (C.1) is not necessary, since every density value corresponds to a physically realizable thickness. Specifications for the LS-DYNA simulation model are shown in Tab. 6.3. Detailed material parameters for the piecewise linear elastic-plastic material model are given in App. C.

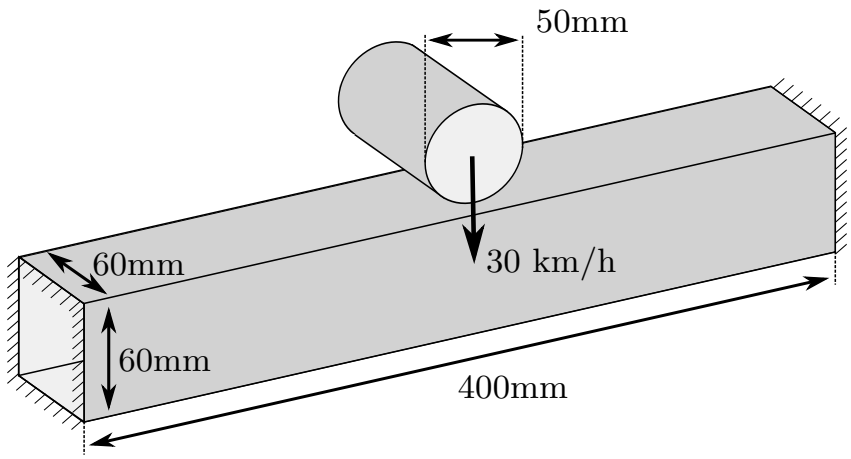
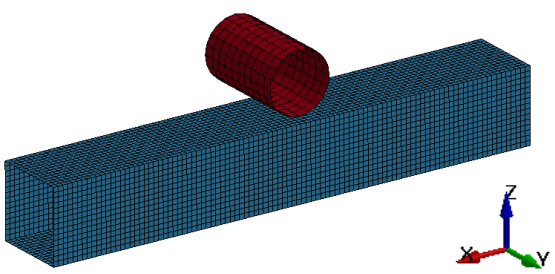


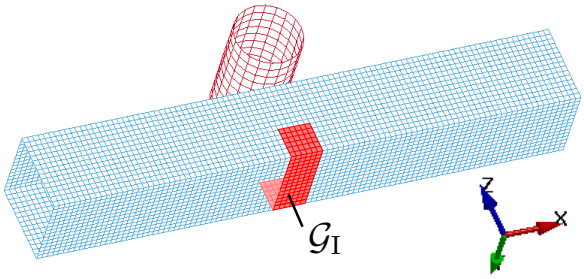
Figure 6.9: The model of frame subject to the crash with a rigid pole.

Table 6.3: Specifications for clamped frame subject to rigid pole crash.

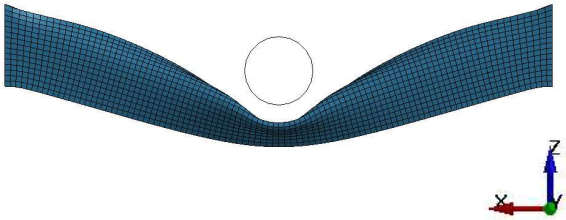
	Symbol	Value
Pole initial z-velocity	-	30km/h
Pole kinetic energy	-	3.5kJ
Number of elements	N	$15 \times 100 \times 4 = 6{,}000$
Termination time	t_{end}	40.0ms
Young's modulus	E_0	70,000MPa
Volumetric mass density	ϱ_0'	$2.7 \cdot 10^{-9}\text{t/mm}^3$
Poisson's ratio	ν	0.33
Yield stress	σ_{y0}	180MPa
Static/Dynamic friction coefficient	-	0.2



(a) Frame and pole.



(b) Highlighting node set $\mathcal{G}_\mathrm{I}$.



(c) Deformed initial structure.

Figure 6.10: The finite element model of the frame subject to the impact of a rigid pole.

Figure 6.10**(a)** shows the LS-DYNA frame model, Fig. 6.10**(b)** shows (in red) the area where the displacement of the nodes should be minimized and Fig. 6.10**(c)** shows the deformation of the initial (uniform thickness) frame at the end of the simulation.

We formulate the optimization problem as:

$$\begin{aligned} \min_{\boldsymbol{\rho}} \mathrm{I}_{\mathcal{G}_\mathrm{I}}(\boldsymbol{\rho}) &= \sqrt{\frac{1}{196}\sum_{i\in\mathcal{G}_\mathrm{I}}(\min_t u_{\mathrm{z}i}(t))^2} \\ \text{s.t.}: \mathbf{r}(t,\boldsymbol{\rho}) &= \mathbf{0}, \\ V(\boldsymbol{\rho})/V_0 &= f, \\ \rho_{\min} \le \rho_i &\le 1,\ i=1,\ldots,N\ , \end{aligned} \tag{6.2}$$

with time step t, nodal displacement $u_{\mathrm{z}i}(t)$ towards the interior and the intrusion measure[4] $\mathrm{I}_{\mathcal{G}_\mathrm{I}}(\boldsymbol{\rho})$. Figure 6.10**(b)** highlights the area in which the node indices belong to set $\mathcal{G}_\mathrm{I}$. The intrusion is quantified by the squared

[4]The intrusion measure in (6.2) quantifies deformation of the absorbing component towards an conceivable inside, and is, therefore, slightly differently defined from Fig.

maximum of the displacement of the nodes within a rectangular area at the centre of the bottom side of the frame, consisting of 196 nodes. Here, we explicitly target intrusion at a defined location, which is not identical to the location of the impact. The state equation is represented by the residual of the dynamic finite element analysis $\mathbf{r}(t, \boldsymbol{\rho})$. The design variables ρ_i scales the thickness of the elements, which can be between 1mm and 10mm.

An important part of the state of an element is reflected by the amount of elastic and plastic deformation. We chose the maximum of the internal energy density $\mathrm{IED}_i(t)$ absorbed by an element as LSF, resulting in the LSF vector:

$$\mathbf{s}_i = \begin{pmatrix} \rho_i \\ \max_t(\mathrm{IED}_i(t)) \end{pmatrix} .$$

For the considered problem, sensitivities are not available, hence gradient-based approaches cannot be applied. This fact justifies the usage of TOPS for the considered problem. Table 6.4 shows the optimization specifications, where the symbols $\rho_{\min}$, f, p were introduced in Sec. 2.2, λ, μ, in Sec. 2.3.1, $r_{\min}$, m in Sec. 4.2, $\Delta F_{\min}$ in Sec. 4.2.1, P in Sec. 4.3.2 and q in App. C respectively, and $\sigma^{(k=0)}$ is the initial step size of the CMA-ES. Important parameters are the numbers for the allowed evaluations M and $M^{\max}$, which were introduced in Sec. 4.2.1. For learning method and parameter choices, similar considerations as for the previous experiment in Sec. 6.2.1 are applicable. In this experiment, we use the maximum of evaluations in each iteration to obtain the best possible update-signal model with this computational budget. The optimization is run for nine different random seeds.

Results from the previous section (and the literature, e.g. [16, 17]), show that HCA is an efficient heuristic to maximize for instance energy absorption or energy uniformness. Yet it is more often seen as a general heuristic for crashworthiness optimization and due to the lack of a dedicated method for the optimization problem in (6.2), it can happen that industrial users fall back to existing methods such as HCA. However, the considered problem relates to cabin safety, for which little intrusion as possible is targeted at defined parts of a structure. This raises the question whether the HCA heuristic is still applicable as it does not adapt to the changed problem. Hence, HCA is applied as baseline for comparisons.

Hence, the specific objective function cannot be addressed easily by the

6.8, where the intrusion is illustrated from the contact point of the the vehicle body towards the inside.

state-of-the-art approaches, due to their specialized nature. Therefore, the generic topology optimization approach provides a unique optimization approach. The results are described in the next section based on the same implementation as described in Sec. 6.2.1.

6.3.2 Optimization Results

Figure 6.11 shows the objective value of all runs versus iterations, respectively evaluations. Again, a monotonic improvement of the objective function is achieved over iterations. The best run performs slightly better than average, it performs a few more iterations before the stopping criterion is fulfilled. In most iterations, the update-signal model is retrained, indicating that the PCM serves rather as adaptive representation than as a sensitivity model. However, it is reused at least once for every run. The best run shows an almost ideal behaviour: the first update-signal model that is trained is able to reduce the objective from 44.26 to 33.19 which is about 74% of the overall improvement. This intermediate result is achieved at the cost of 600 evaluations only.

The best optimized thickness distribution obtained by TOPS-PCM is shown in Fig. 6.12**(a)**. The areas along the top edges of the frame are increased in thickness, especially close to the supports. The lower side of the frame is driven to the minimum thickness. Near the impact edges the

Table 6.4: Specifications of PCM-TOPS for frame intrusion minimization.

	Symbol	Value
Evaluations per learning step	M	600
Max. evaluations per iteration	$M^{\max}$	600
Number of prototypes	P	15
Improvement threshold	$\Delta F_{\min}$	0.001
Target volume fraction	f	0.2
Filter radius	$r_{\min}$	6.0mm
Number of offspring	λ	12
Number of parents	μ	6
Minimum density	$\rho_{\min}$	0.10
Move limit	m	0.1
Initial global step size	$\sigma^{(k=0)}$	0.3

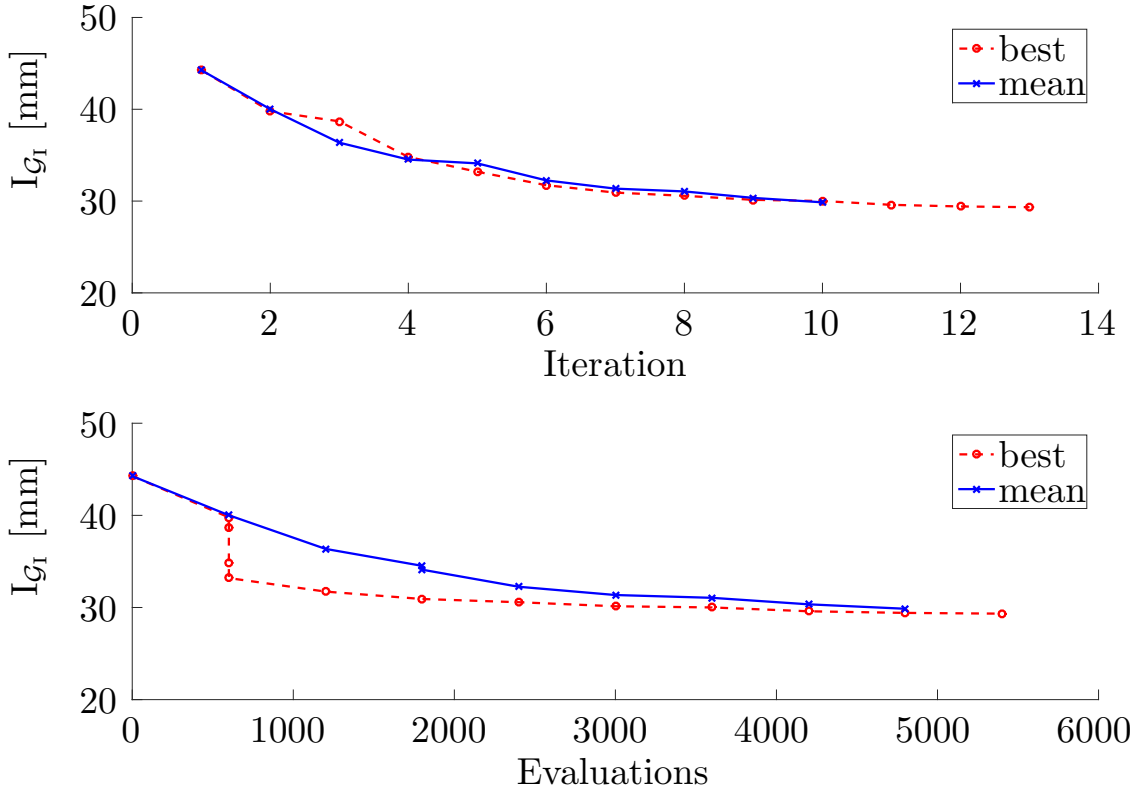
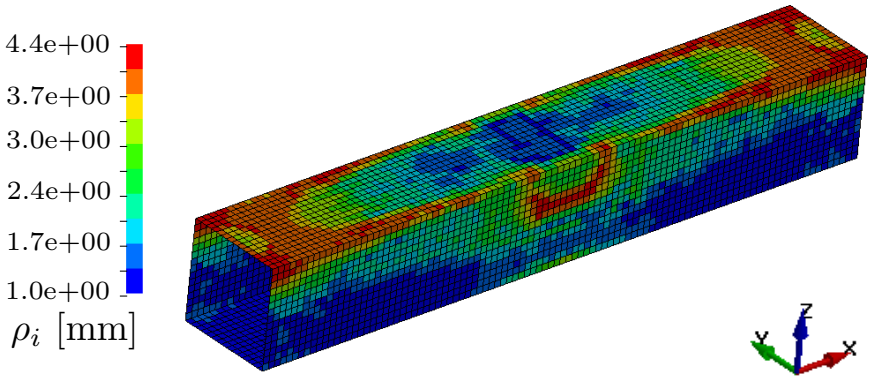


Figure 6.11: Intrusion versus the number of iterations (top), respectively evaluations (bottom) for frame intrusion minimization with TOPS-PCM, shown for the best and the mean run.

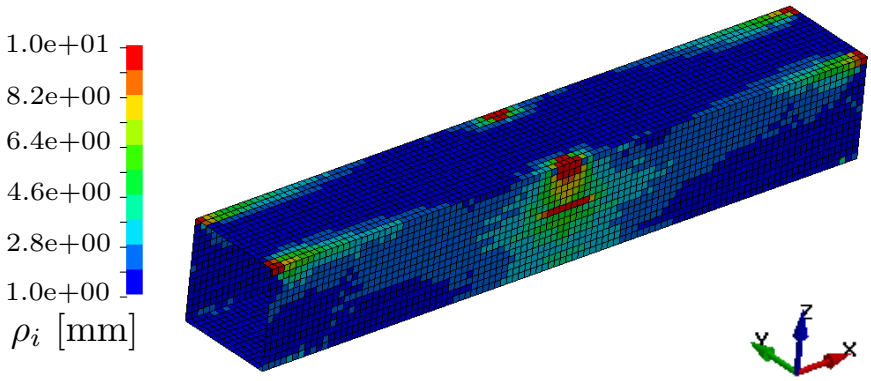
structure is as well reinforced. Interestingly, the full thickness is not utilized. On average, a total of roughly 4,400 evaluations during ten iterations is required.

The HCA baseline algorithm was stopped after 200 iterations (and 200 evaluations accordingly) when the algorithm specific mass redistribution convergence criterion was clearly oscillating periodically. The low number of evaluations is an advantage of the HCA method. A different thickness distribution than for PCM-TOPS is obtained, compare Fig. 6.12**(b)**.

The deformations of the PCM-TOPS structure and the HCA baseline structure are shown in Fig. 6.13 and 6.14. Figure 6.13 shows the deformations of both structures from a frontal perspective. For the HCA design, the overall response is stiffer and the complete frame is in a bending mode. Accordingly, the frame itself is only slightly deformed at the impact location. However, the bottom side of the frame is intruding significantly at the location of the nodes that are subject to the objective function in (6.2). The PCM-TOPS structure shows a rather compliant behaviour at the im-



(a) Optimized result, PCM-TOPS.



(b) Optimized result, HCA.

Figure 6.12: Optimized frame thickness distribution subject to intrusion minimization by PCM-TOPS **(a)** and by HCA **(b)**, colour-coded by the elemental thickness in mm.

pact location caused by a folding of the vertical walls. The impactor is pushing the upper wall into the interior of the *frame*. However, the lower wall and the associated nodes are hardly moving in the (hypothetical) interior of the *vehicle*. With respect to the optimization formulation, PCM-TOPS achieves a lower intrusion measure and, hence, a better optimization result than HCA. Rotating the model towards the contemplator, yields the second perspective in Fig. 6.14. Here, the relevant intrusion is even more clearly visible. Furthermore, the smoothness of the resulting deformations might be interesting as well. From a safety perspective the PCM-TOPS result is favourable, since the sharp edges of the baseline result could bear potential risk for injuries.

Figure 6.15 shows the average z-displacement of the node set $\overline{u}_z(t) =$

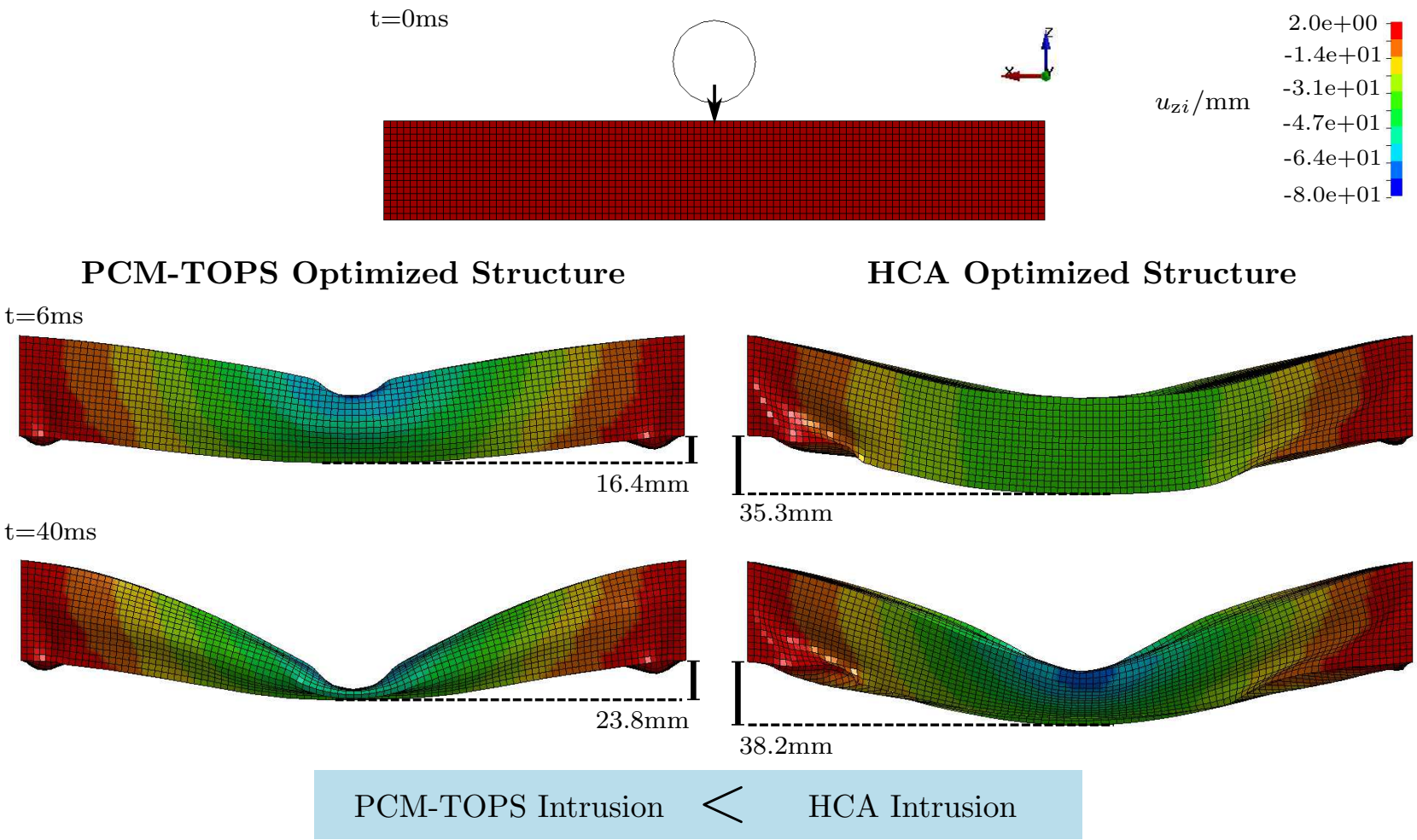


Figure 6.13: Deformation at 6ms and 40ms (final simulation time step) of the optimized frames subject to intrusion minimization by PCM-TOPS and by HCA, seen from the front, colour-coded by the displacement in z-direction in mm.

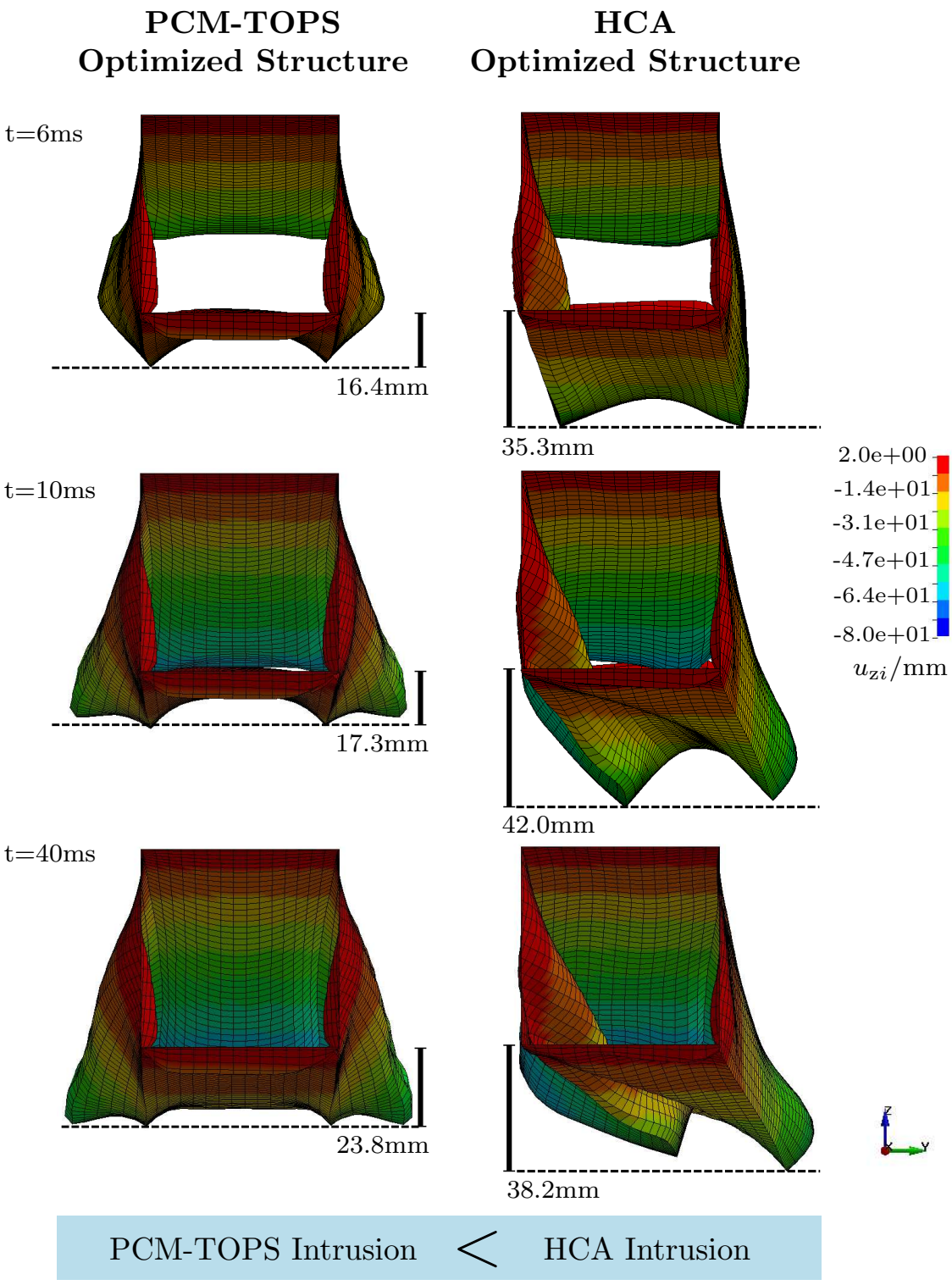


Figure 6.14: Deformation at 6ms, 10ms, and 40ms (final simulation time step) of the optimized frames subject to intrusion minimization by PCM-TOPS and by HCA, seen from the side, colour-coded by the displacement in z-direction in mm.

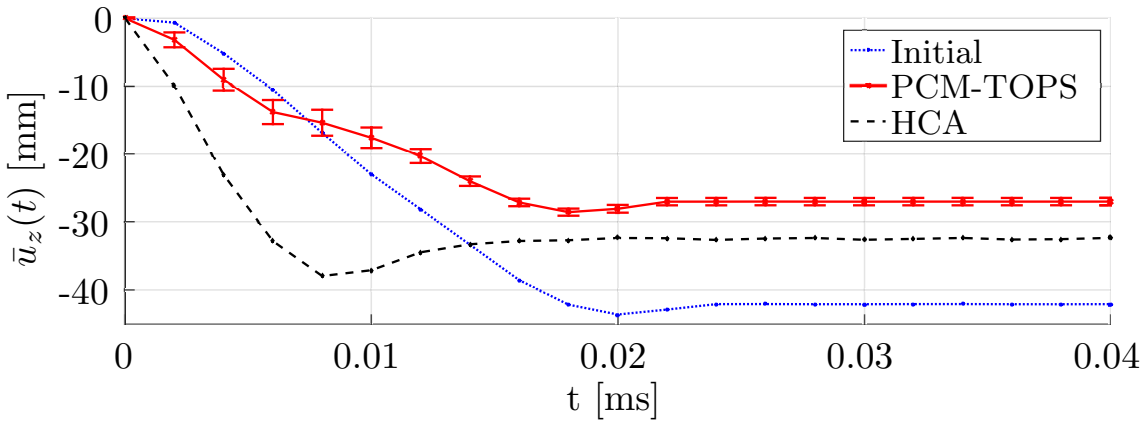


Figure 6.15: The average z-displacement $\bar{u}_z(t)$ of the intruding node set $\mathcal{G}_\mathrm{I}$ during the crash, of the frame subject to intrusion minimization for the initial and the optimized designs by PCM-TOPS and HCA. For PCM-TOPS, the mean of the runs is plotted with bars indicating the standard deviation.

$\frac{1}{196}\sum_{i\in\mathcal{G}_\mathrm{I}} u_{zi}(t)$. The result from PCM-TOPS has significantly lower average displacement for most simulation time steps, compared to the initial uniform thickness structure and the HCA structure.

In order to improve our understanding of the PCM-TOPS, we study the evolved update-signal models in more detail. The empirical correlation coefficient between the update-signals and the LSFs is shown in Fig. 6.16 for the best run (More information and the coefficient's definition are given in App. A.4. For the density, there is an unsteady but weak to moderate anti-correlation. This can be interpreted intuitively as the tendency to more often increase the density of elements that have a higher density. Thus, those have higher (in absolute terms) update-signals. For the internal energy density, there is only a very weak anti-correlation, except for the first four iterations, where the anti-correlation is slightly stronger with a value around -0.4. Hence, in these first iterations, where the update-signal model is reused (compare Fig. 6.11), the internal energy density LSF provides a good estimate where to add and remove material. Later on, the correlation tends towards 0, as a new model is trained every iteration, and the update-signal model takes the form of an indirect representation. In this phase, the internal energy density is not as useful for modelling the sensitivity of the intrusion objective function in (6.2) as it is at the beginning. Yet, this can be compensated by the evolutionary search.

To examine the working principle, the optimized update-signals are mapped to the structure for the different iterations, see Fig. 6.17. Again,

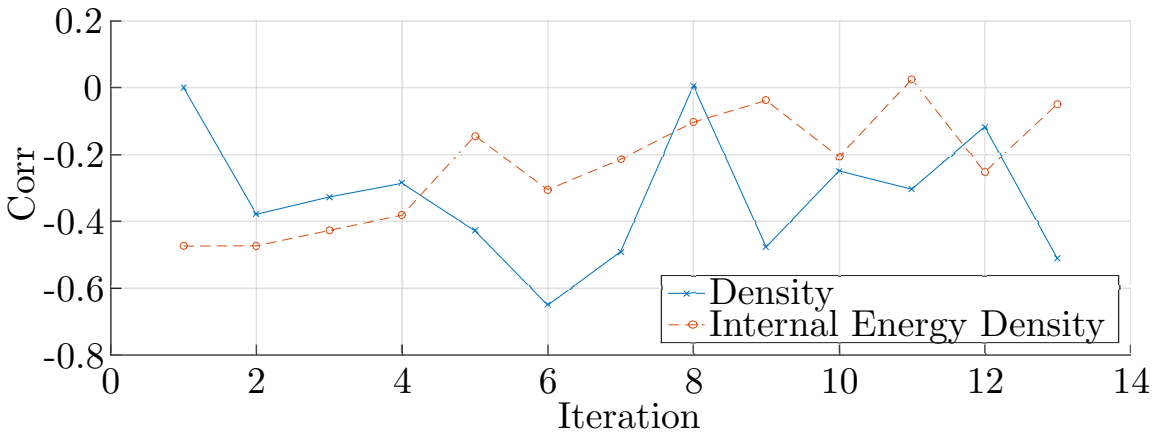


Figure 6.16: The empirical correlation coefficient between update-signals and LSFs plotted over the iterations for the frame structure optimization. A very weak to weak anti-correlation between LSFs and update-signals is observed. Since a minimization is considered, a perfect anti-correlation of -1 would imply that the higher the value of the LSF the larger the change of the density and vice versa.

each colour corresponds to one cluster in LSF space. Here, we consider one of the runs where the update-signal model is reused several times during the iterations. The clusters refer to relatively complex topological patterns in the design space that are often disconnected and distributed globally over the design space. Some of the patterns are reappearing in several iterations. The resulting optimized thickness distribution in Fig. 6.12**(a)** can be recognized in the update-signals along the upper side/edges of the frame where the update-signal suggest to add material in early and late iterations, and removal or relative little change of material in iterations 3 to 5. In this run, in iteration 2, the update-signal model of the first iteration is reused. Note how this results in similarities to the preceding iteration in colours and element clusters. Although the update-signal model is the same, the changed LSFs result in a change of closest prototypes and, hence, the update-signals change. Similarly, in iterations 4 and 5 the model from iteration 3 is reused, resulting in similar, yet subtly changed update-signals. This is followed by iterations where a retraining of the PCM model is performed, resulting in highly different update-signals.

In conclusion, the application of the update-signals in PCM-TOPS achieve a better optimization result compared to the HCA. The baseline algorithm HCA cannot adapt its fundamental assumption of a uniform energy distribution to the given objective function. This leads to

an overly stiff structure, which is a suboptimal result for the optimization target. The generic component of PCM-TOPS is a conceptual advantage compared to the specialized heuristic. The optimized thickness distribution results in a folding of the frame that avoids the undesired intrusion. This demonstrates on a first application-oriented problem that the generic method proposed in this thesis in fact clears the way to novel optimization problems in the domain of crashworthiness.

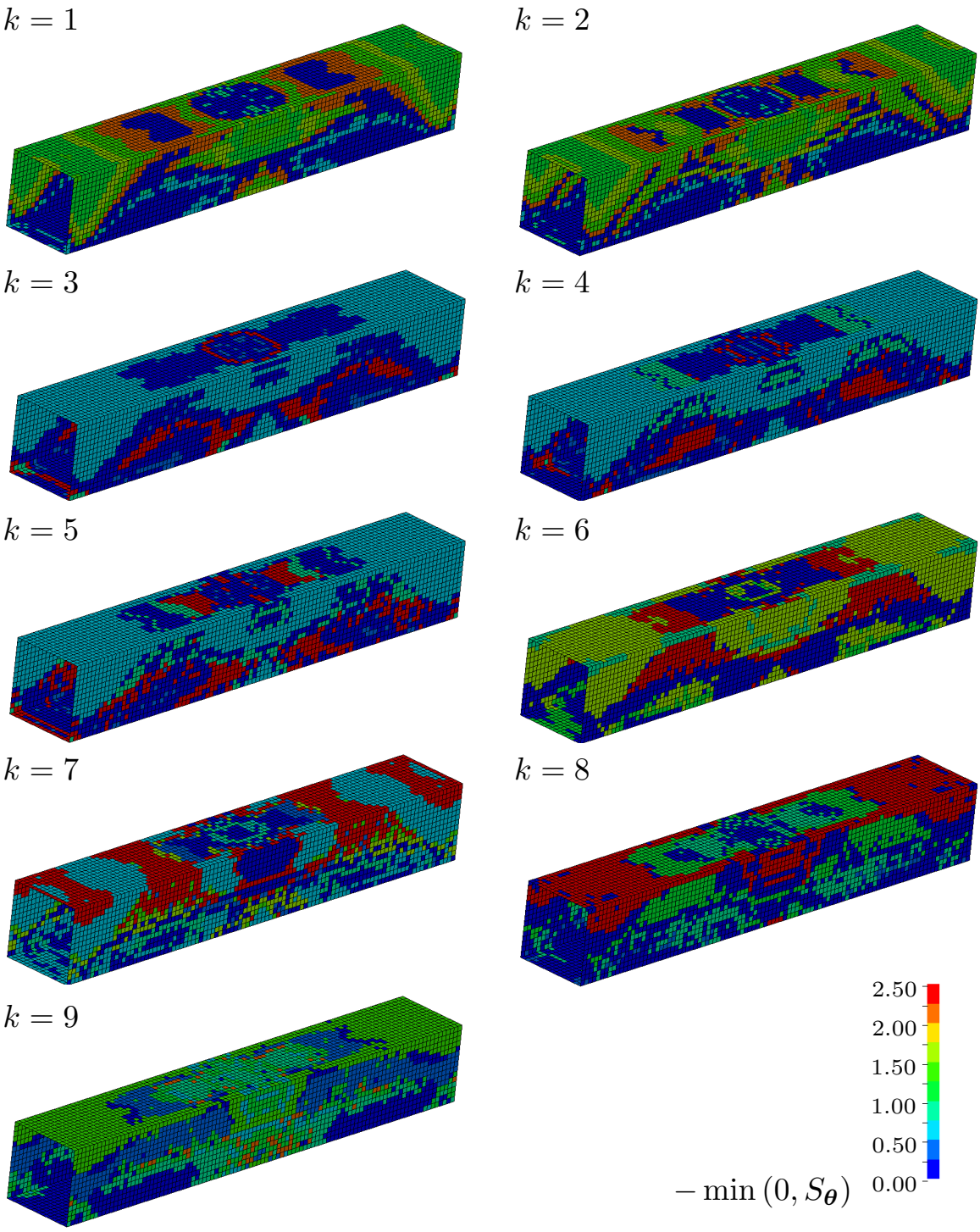


Figure 6.17: The optimized update-signals colour-coded onto the frame structure for different iterations, where blue is a signal for reduction and red for increase of the thickness, respectively.

6.4 Summary of Contribution

This chapter demonstrates the applicability and usefulness of the proposed generic approach for non-linear crashworthiness topology optimization. Topology Optimization by Predicting Sensitivities with Piecewise-Constant Model (PCM-TOPS) is applied to two crash problems and compared to baselines obtained by HCA. The first experiment on maximizing energy absorption of a two-dimensional beam considers a typical HCA use case. Although the baseline maintains a higher performance in terms of energy absorption in this set-up, PCM-TOPS nevertheless results in a set of meaningful and interesting design concepts with high energy absorption capabilities. This is achieved without the previous assumption on uniform energy distribution in the optimized structure. In the second problem, an intrusion measure of a frame structure subject to crash is minimized. Although no additional assumptions are is introduced, PCM-TOPS generalizes to this different objective function and obtains a well-performing optimization result. The optimized structure shows a different deformation behaviour and clearly outperforms the baseline. This emphasizes the advantage of the proposed generic method. Concluding, the chapter shows that the generic topology optimization approach can be transferred and successfully applied to the challenging domain of crashworthiness topology optimization. From a more general perspective, this demonstrates that the method is able to address the optimization of novel, so far unchallenged topology optimization problem.

7 Conclusions

7.1 Main Results

This thesis proposes topology optimization based on integration of learning concepts from the field of computational intelligence. Inspiration came from the field of artificial, evolutionary development in which fascinating biological models for the growth of structures have been studied. Some of these models focused on evolving cellular behaviours that result in optimized structures [182, 183]. Interestingly, also classical topology optimization algorithms show aspects of a developmental process, due to their pronounced similarity to cellular automata. However, computational steps of these topology optimization algorithms are performed based on a mathematical perspective, instead of a biological motivation. In this sense, we can recognize similarity between the usage of mathematical gradients for topology optimization and a developmental process that is influenced by environmental information. This similarity clarifies that the task of a biologically-inspired structural growth is strongly related to modelling the gradient of the objective function. Assuming a suitable learning process, we receive a generic topology optimization method that is more general than classical approaches. This novel optimization method is the main research contribution of this thesis. The individual steps are summarized in the following paragraphs.

An eminent difficulty for general, stochastic optimization of continuum topology optimization problems resides in the design freedom naturally reflected by an accordingly high dimensionality of the search space. Evolutionary computation literature provides plenty of different solution representations that more or less address this challenge. For the first time since the latest survey about a decade ago [101], this thesis collects and structures the approaches found in the literature. A categorization into three classes is proposed: grid, geometric and indirect representations. While grid representations are limited to small problems, geometric representations can provide a balance between design complexity and search space

dimensionality. The potential to represent a large variety of topological designs is identified for indirect representations, although they are rarely studied in relation to topology optimization. The presented overview helps to realize that there are almost no concepts for the usage of additional simulation data in the evolutionary computation literature. This striking absence was one of the motivations for deriving a novel approach that is able to systematically utilize this usually neglected information.

In general, the fields of topology optimization and evolutionary computation have different conceptions of how much problem-specific knowledge is available. For classical topology optimization, typically in-depth, physical understanding and mathematically formulated derivatives are a prerequisite. In contrast, the optimization progress of evolutionary algorithms is guided by objective function values, i.e. it is intrinsically data-driven. This thesis proposes a novel, generic method that enables a combination of data- and knowledge-driven aspects. The data-driven aspect is realized by a learning component that harvests structural simulation data. This data implicitly contains information on gradients and can, if available, be refined by expert knowledge. In the developed method, the available simulation data is captured by the concept of Local State Features (LSF). Structural updates during the optimization are performed by a learned model that provides a search direction depending on the LSFs.

Evolutionary algorithms can be utilized for the optimization of the corresponding model parameters, since the model complexity is independent of the high number of structural design variables. The optimized model then, provides update-signals that improve the structure. Since the sensitivity of the objective function with respect to the design variables is identified as the primary learning goal, the method is termed Topology Optimization by Predicting Sensitivities (TOPS). This clear goal makes it possible to address the problem, as well, with a supervised learning approach that is based on finite difference sampling. A primary contribution of the presented work is the elaboration of the learning methods, which lead to the following modelling approaches:

- A feed-forward artificial neural network processing LSFs where weights and biases are tuned by CMA-ES (NE-TOPS).
- A piecewise-constant model that is based grouping elements with similar LSFs around prototypes (PCM-TOPS). For all elements assigned to one prototype, a constant update-signal is assumed and tuned by CMA-ES.

- Supervised learning based on linear or support vector regression (LIN/SVR-TOPS).

Empirical studies on the minimum compliance problem demonstrate the feasibility of the approaches by reproduction of a reference structure. As can be expected, the results depend to a large part on the choice of LSFs and, hence, the selected amount of previous knowledge. Depending on the relation of the LSFs to the design variables' sensitivities, the effort for learning may be reduced, until in an extreme case no learning is required and TOPS reduces to a standard gradient-based method. Although the approach is different from classical evolutionary algorithms, it is interesting to realize that optimizing the proposed piecewise-constant model is equivalent to optimizing an adaptive state-based representation that groups elements with similar LSFs. This establishes a direct link between performing a gradient-based search and a stochastic one, in which the latter uses an adaptive, indirect representation.

In the minimum compliance problem, for which finite difference sampling is possible, training a supervised regression model for the update-signals is more accurate and computationally cheaper in terms of evaluations compared to the evolutionary approach. Support vector regression has shown to be a reliable predictor, which is able to adequately replace the sensitivities for various combinations of LSFs. However, for the supervised learning of the update-signal model the applicability is limited to objective functions with smooth responses, due to the necessity of finite difference samples. In other cases, in which the sensitivity information in the LSFs is very implicit, or gradient-based optimization steps do not suit to improve the design, TOPS can succeed with the methods of variation and selection in one of the two evolutionary learning approaches. Evolutionary optimization does not need to follow the gradient explicitly and can provide any update-signals that improve the structure. However, evolutionary optimization requires a high amount of function evaluations, currently reducing its applicability to problems with small to medium-sized simulation models. For the reference case, a piecewise-constant approximation is superior to a smooth neural network approximation when the relation between LSFs and sensitivities is non-linear. When the task is mainly selection of a LSF, a simple neural network is more appropriate. The presented comprehensive evaluation demonstrates the feasibility of TOPS and identifies assets and limitations of the various, developed learning methods.

In order to demonstrate practical usefulness, PCM-TOPS was trans-

ferred to the domain of crashworthiness topology optimization, where sensitivity information is usually not available. Crashworthiness topology optimization is especially interesting due to its relevance for the vehicle design process in the automotive industry. In comparison with a domain-specific heuristic baseline, TOPS achieves interesting structural concepts with qualitatively similar performance when considering energy absorption maximization. The adaptability of TOPS is highlighted with a thin-walled frame example, for which an intrusion measure is minimized. The result is a superior solution with a distinct deformation behaviour compared to the baseline. In particular, the intrusion problem represents a category of practical optimization problems, for which optimization is basically not realizable without the proposed method. General methods cannot handle the number of design variables, and specialized methods are impeded by the problem complexity. This step towards application demonstrates the ability of TOPS to address challenging topology optimization problems for which otherwise no suitable method exists.

Hence, the main result of this thesis is a method for addressing complex conceptual design problems, for which classical methods are not feasible. Methods of computational intelligence, respectively statistical modelling, are utilized for topology optimization that is able to adapt to the considered problem. A model is implemented by supervised or evolutionary learning that provides updates for improving the design. This generic search direction renders the derivation of mathematical sensitivities unnecessary, which so far requires human experts. Although many directions for refinement remain and overall more comparisons of methods are required, the learning process is shown to be successful. The applicability is demonstrated on a practical problem from the domain of crashworthiness, where promising results are obtained. The empirical evidence shows the usefulness of the approach for instance for the automotive industry, where generic topology optimization may lead to more creative and efficient design solutions in the future.

7.2 Future Directions

There are large commonalities between the utilized, density-based, and other approaches to the topology optimization problem. The majority of the established methods relies on mathematical derivatives. Although the formulations take different forms, it has been observed by Sigmund and

Maute [171] that the sensitivity information is essentially the same in different approaches. For instance, the "Sensitivity Number" used in discrete BESO approaches corresponds to the sensitivity from density-based methods. Topological derivative methods use precise, analytic information on introducing an infinitesimal small hole in the structure. However, practically, a finite element is removed from a mesh, which is not infinitely small and, hence, the sensitivity relates to that of the foregoing methods. Also, for level set methods, sensitivities on the boundary of the level set function can be related to the sensitivity of the density-based methods on the structural boundary. Hence, the underlying learning goal of the generic approach is fairly independent of the density-based problem formulation and transferable to the mentioned approaches. Concretely, the update-signals generated by the model could be used, for instance, to remove and admit complete elements, increase or decrease the thickness of cross sections or substitute surface sensitivities of a level set function.

Another important aspect for future research are techniques to select LSFs. So far, it is assumed that LSFs are selected by the user, either based on intuition or based on a simplified analysis. However, especially for problems with time-dependent simulations, there can be numerous candidate LSFs and intuition might be limited, hence, an automatized choice would be favourable. This could be addressed by feature selection or dimensionality reduction techniques. The latter is mainly interesting for reducing computational cost of evolutionary learning, since fewer LSFs imply a reduction of search space dimensionality. It also seems promising to investigate the determination of elements for sensitivity-sampling by suitable design of experiments methods, instead of random schemes. For the proposed piecewise-constant model a systematic clustering method for the determination of element prototypes has potential to improve the method, as well.

The research direction that possibly promises the highest impact on the real-world design process is the continued application of TOPS to realistic use cases. Crashworthiness topology optimization is the distinguished example in this thesis, providing a starting point for future work. However, the considered crash problems were merely examples of what could be addressed. Today, many crash application challenges remain, for instance extruded vehicle body components or energy-absorbing, progressively-folding tubes. Since every application has its pitfalls and peculiarities, it is reasonable to dedicate further research to a specific domain. For this domain, specialized LSFs could be identified based on simplified models, for instance, based on Equivalent Static Load models as in [35].

A Theory

A.1 Adjoint Sensitivities

As an example for adjoint sensitivity analysis, let us consider the minimum compliance problem:

$$\begin{aligned} \min_{\boldsymbol{\rho}} c(\boldsymbol{\rho}) &= \mathbf{l}^{\mathrm{T}}\mathbf{u} \\ \text{s.t. : } & \mathbf{K}(\boldsymbol{\rho})\mathbf{u} = \mathbf{l}, \\ & V(\boldsymbol{\rho})/V_0 = f, \\ & 0 < \rho_{\min} \leq \rho_i \leq 1,\ i = 1, \ldots, N \ , \end{aligned}$$

with stiffness matrix $\mathbf{K}$, displacement field $\mathbf{u}$ and load vector $\mathbf{l}$.

The sensitivity $\partial c(\boldsymbol{\rho})/\partial \rho_i$ can be derived by adding the equilibrium equation as zero function [30]:

$$c(\boldsymbol{\rho}) = \mathbf{l}^{\mathrm{T}}\mathbf{u} - \mathbf{v}^{\mathrm{T}}(\mathbf{K}(\boldsymbol{\rho})\mathbf{u} - \mathbf{l}) \ ,$$

with an arbitrary but fixed vector $\mathbf{v}$. Derivation yields:

$$\frac{\partial c}{\partial \rho_i} = \frac{\partial \mathbf{l}^{\mathrm{T}}}{\partial \rho_i}\mathbf{u} + \mathbf{l}^{\mathrm{T}}\frac{\partial \mathbf{u}}{\partial \rho_i} - \mathbf{v}^{\mathrm{T}}\left(\frac{\partial \mathbf{K}}{\partial \rho_i}\mathbf{u} + \mathbf{K}\frac{\partial \mathbf{u}}{\partial \rho_i} - \frac{\partial \mathbf{l}}{\partial \rho_i}\right) \ .$$

Since the load is independent of the design, i.e. $\frac{\partial \mathbf{l}}{\partial \rho_i} = 0$, we rearrange:

$$\begin{aligned} \frac{\partial c}{\partial \rho_i} &= \mathbf{l}^{\mathrm{T}}\frac{\partial \mathbf{u}}{\partial \rho_i} - \mathbf{v}^{\mathrm{T}}\frac{\partial \mathbf{K}}{\partial \rho_i}\mathbf{u} - \mathbf{v}^{\mathrm{T}}\mathbf{K}\frac{\partial \mathbf{u}}{\partial \rho_i} \\ &= -\mathbf{v}^{\mathrm{T}}\frac{\partial \mathbf{K}}{\partial \rho_i}\mathbf{u} + \left(\mathbf{l}^{\mathrm{T}} - \mathbf{v}^{\mathrm{T}}\mathbf{K}\right)\frac{\partial \mathbf{u}}{\partial \rho_i}. \end{aligned}$$

From this we have

$$\frac{\partial c}{\partial \rho_i} = -\mathbf{v}^{\mathrm{T}}\frac{\partial \mathbf{K}}{\partial \rho_i}\mathbf{u} \ ,$$

when $\mathbf{v}$ satisfies the adjoint equation

$$\mathbf{l}^{\mathrm{T}} - \mathbf{v}^{\mathrm{T}}\mathbf{K} = 0 \ ,$$

which is the case for $\mathbf{v} = \mathbf{u}$. Therefore, in case of the compliance problem no additional computations are required for the adjoint state.

Usually solving the adjoint equation requires additional computations, as we can see when we consider a different objective function. Let us consider minimization of displacement u_t of an arbitrary target node t instead of the compliance:

$$\begin{aligned} \min_{\boldsymbol{\rho}} u_t(\boldsymbol{\rho}) &= \mathbf{l}_t^\mathrm{T}\mathbf{u} \\ \text{s.t. : } & \mathbf{K}(\boldsymbol{\rho})\mathbf{u} = \mathbf{l}, \\ & V(\boldsymbol{\rho})/V_0 = f, \\ & 0 < \rho_\mathrm{min} \leq \rho_i \leq 1,\ i = 1, \ldots, N \ . \end{aligned}$$

Then, the displacement u_t of the target node t can be determined with a virtual load vector $\mathbf{l}_t$ that is unity at index t and zero otherwise.

For the sensitivity analysis, we follow an analogous derivation as for the compliance sensitivities. The sensitivity of the target displacement component with respect to a design variable i is obtained by adding the zero function:

$$u_t = \mathbf{l}_t^\mathrm{T}\mathbf{u} - \mathbf{v}^\mathrm{T}\left(\mathbf{K}\mathbf{u} - \mathbf{l}\right) \ ,$$

with the arbitrary but fixed vector $\mathbf{v}^\mathrm{T}$ and differentiation:

$$\begin{aligned} \frac{\partial u_t}{\partial \rho_i} &= \frac{\partial \mathbf{l}_t^\mathrm{T}}{\partial \rho_i}\mathbf{u} + \mathbf{l}_t^\mathrm{T}\frac{\partial \mathbf{u}}{\partial \rho_i} - \mathbf{v}^\mathrm{T}\left(\frac{\partial \mathbf{K}}{\partial \rho_i}\mathbf{u} + \mathbf{K}\frac{\partial \mathbf{u}}{\partial \rho_i} - \frac{\partial \mathbf{l}}{\partial \rho_i}\right) \\ &= \mathbf{l}_t^\mathrm{T}\frac{\partial \mathbf{u}}{\partial \rho_i} - \mathbf{v}^\mathrm{T}\frac{\partial \mathbf{K}}{\partial \rho_i}\mathbf{u} - \mathbf{v}^\mathrm{T}\mathbf{K}\frac{\partial \mathbf{u}}{\partial \rho_i} \\ &= -\mathbf{v}^\mathrm{T}\frac{\partial \mathbf{K}}{\partial \rho_i}\mathbf{u} + \left(\mathbf{l}_t^\mathrm{T} - \mathbf{v}^\mathrm{T}\mathbf{K}\right)\frac{\partial \mathbf{u}}{\partial \rho_i}. \end{aligned}$$

When the vector $\mathbf{v}$ solves the adjoint equation we obtain:

$$\frac{\partial u_\mathrm{t}}{\partial \rho_i} = -\mathbf{v}_t^\mathrm{T}\frac{\partial \mathbf{K}}{\partial \rho_i}\mathbf{u} \ .$$

Here, the adjoint equation is:

$$\mathbf{l}_t^\mathrm{T} - \mathbf{v}^\mathrm{T}\mathbf{K} = \mathbf{0} \ .$$

As it has the form of an equilibrium equation, it can be intuitively understood as applying an adjoint (dummy) load vector $\mathbf{l}_t^\mathrm{T}$ to the design space, that yields the adjoint state $\mathbf{v}$. Hence, unless the point of the load $\mathbf{l}$ is identical to the target node t, $\mathbf{v} \neq \mathbf{u}$ and solving the adjoint equation requires an additional finite element computation.

A.2 Sensitivity Regression Models

A.2.1 Linear Regression

At first a basic linear regression model is considered. Linear regression is commonly used in engineering, often as response surface surrogates, its parameters express linear relations of the features to the target. Linear models are intuitive to understand and computationally fast to train. By using higher order variants of the original features also multivariate polynomial models are captured by linear regression models.

A linear regression model is defined as in (4.13):

$$S_{\boldsymbol{\theta}}(\mathbf{s}) = \theta_0 + \theta_1 s_1 + ... + \theta_J s_J = \boldsymbol{\theta}^T \begin{bmatrix} 1 \\ \mathbf{s} \end{bmatrix} \ ,$$

with the model parameters $\boldsymbol{\theta}$ and the J-dimensional feature vector $\mathbf{s}$. The model parameters are determined by minimizing the cost function

$$\text{Cost} = 1/(2T) \sum_{t=1}^{T} \left(S_{\boldsymbol{\theta}}(\mathbf{s}_t) - y_t\right)^2 + \lambda \sum_{j=0}^{J} \theta_j^2 \ , \tag{A.1}$$

with a set of T training samples ($\mathbf{s}_t$,y_t) and a regularization parameter λ. The second term is a typical regularization term, that is included to avoid over-fitting the training data, since for $\lambda = 0$ the problem is ill-posed. The specific choice of regularization is termed ridge regression [83], and ensures flatness of the function by penalizing large values of the model parameters, such that overall variance and especially the influence of unnecessary input features is reduced. The values for θ are found by solving the least squares problem posed by minimization of (A.1). A good value for the regularization parameter can be found by a line search using cross-validation.

A.2.2 Support Vector Regression

For prediction, Support Vector Machines keep a subset of the training samples as support vectors and apply the so-called kernel trick to transform the input features space to a higher dimensional space.

The concept of support vectors can be used for regression tasks, i.e. Support Vector Regression (SVR) [173, 194]:

$$S_{\boldsymbol{\theta}}(\mathbf{s}) = \sum_{l=1}^{\hat{L}} \theta_l \hat{k}\left(\mathbf{s}_l,\mathbf{s}\right) + \theta_0 \ ,$$

where $\mathbf{s}_l$ are the $\hat{L}$ support vectors and $\hat{k}(\mathbf{s}_l,\mathbf{s})$ is the kernel function, representing a dot product of a transformation of the input vector in a higher dimensional feature space. Only the dot product is relevant, such that the transformation does not need to be known explicitly. It is sufficient to show that a kernel is admissible, i.e. it exists a transformation that the kernel can be represented as a dot product in the feature space. An admissible kernel, that is commonly seen as a good choice, is the Radial-Basis-Function (RBF) kernel, with $\hat{k}(\mathbf{s},\mathbf{s}') = \exp^{-\gamma\|\mathbf{s}-\mathbf{s}'\|^2}$. The size of the RBF is controlled by γ. In contrast to linear kernels, the RBF kernel maps the input vector to a higher dimensional feature space and therefore can handle the case of non-linear relations between input and target. The support vectors are a subset of the data samples, that need to be stored and are required for prediction.

The target of ϵ-SVR [194] is a prediction error smaller than ϵ:

$$\begin{aligned} &\min \left[\frac{1}{2}||\mathbf{w}||^2 + C\sum_i^T(\xi_i + \xi_i^*)\right] \\ &\text{s.t.: } y_i - \langle \mathbf{w}, \mathbf{s}_i\rangle - b \leq \epsilon + \xi_i, \\ &\qquad \langle \mathbf{w}, \mathbf{s}_i\rangle + b - y_i \leq \epsilon + \xi_i^*, \\ &\qquad \xi_i, \xi_i^* \geq 0 \ , \end{aligned} \tag{A.2}$$

where $\mathbf{w}$ are the coefficients of a linear function, and $\langle \cdot \ , \ \cdot \rangle$ is the dot product and $\langle w, w\rangle = ||\mathbf{w}||^2$, and ξ, ξ^* are slack variables introduced to soften the constraints. The constant C determines the trade-off between precision and tolerating prediction errors larger than ϵ, thus it has a very similar regularization function as $1/\lambda$ for linear regression in (A.1). The smaller C the fewer support vectors will be required to keep the error sufficiently low, this means a more sparse solution is found.

Problem (A.2) can be formulated as a quadratic optimization problem that can be solved by constructing a Lagrangian function and introducing dual variables for the constraints. As solver, existing quadratic programming packages can be applied. The found function is the flattest function approximating the target within the error bound (in feature space). It is easy to obtain accurate results with the SVR, when following the recommendations of Hsu et al. [87]. Concretely, it is necessary to follow the steps of scaling, using the RBF kernel, finding the sensitive parameters C and γ by a grid-search on a cross validation data set, and final re-training. A disadvantage is the relatively high computational cost for training of the SVR, however in the considered use case it is small compared to the cost

of gathering the training samples.

A.3 Compliance Topology Optimization

The elemental stiffness matrix used for the compliance reference problem in Sec. 5.1 from [5, 168] is:

$$\mathbf{K}_0 = \frac{E}{(1-\nu^2)} \cdot \begin{pmatrix} \frac{1}{2}-\frac{\nu}{6} & \frac{1}{8}-\frac{\nu}{8} & -\frac{1}{4}-\frac{\nu}{12} & -\frac{1}{8}-\frac{3\nu}{8} \\ \frac{1}{8}-\frac{\nu}{8} & \frac{1}{2}-\frac{\nu}{6} & \frac{1}{8}-\frac{3\nu}{8} & \frac{\nu}{6} \\ -\frac{1}{4}-\frac{\nu}{12} & \frac{1}{8}-\frac{3\nu}{8} & \frac{1}{2}-\frac{\nu}{6} & -\frac{1}{8}-\frac{\nu}{8} \\ -\frac{1}{8}-\frac{3\nu}{8} & \frac{\nu}{6} & -\frac{1}{8}-\frac{\nu}{8} & \frac{1}{2}-\frac{\nu}{6} \\ -\frac{1}{4}+\frac{\nu}{12} & -\frac{1}{8}-\frac{\nu}{8} & \frac{\nu}{6} & \frac{1}{8}-\frac{3\nu}{8} \\ -\frac{1}{8}-\frac{\nu}{8} & -\frac{1}{4}+\frac{\nu}{12} & -\frac{1}{8}-\frac{3\nu}{8} & -\frac{1}{4}-\frac{\nu}{12} \\ \frac{\nu}{6} & -\frac{1}{8}-\frac{3\nu}{8} & -\frac{1}{4}+\frac{\nu}{12} & \frac{1}{8}-\frac{\nu}{8} \\ \frac{1}{8}-\frac{3\nu}{8} & -\frac{1}{4}-\frac{\nu}{12} & \frac{1}{8}-\frac{\nu}{8} & -\frac{1}{4}+\frac{\nu}{12} \end{pmatrix}$$

$$\begin{pmatrix} -\frac{1}{4}+\frac{\nu}{12} & -\frac{1}{8}-\frac{\nu}{8} & \frac{\nu}{6} & \frac{1}{8}-\frac{3\nu}{8} \\ -\frac{1}{8}-\frac{\nu}{8} & -\frac{1}{4}+\frac{\nu}{12} & -\frac{1}{8}-\frac{3\nu}{8} & -\frac{1}{4}-\frac{\nu}{12} \\ \frac{\nu}{6} & -\frac{1}{8}-\frac{3\nu}{8} & -\frac{1}{4}+\frac{\nu}{12} & \frac{1}{8}-\frac{\nu}{8} \\ \frac{1}{8}-\frac{3\nu}{8} & -\frac{1}{4}-\frac{\nu}{12} & \frac{1}{8}-\frac{\nu}{8} & -\frac{1}{4}+\frac{\nu}{12} \\ \frac{1}{2}-\frac{\nu}{6} & \frac{1}{8}-\frac{\nu}{8} & -\frac{1}{4}-\frac{\nu}{12} & -\frac{1}{8}-\frac{3\nu}{8} \\ \frac{1}{8}-\frac{\nu}{8} & \frac{1}{2}-\frac{\nu}{6} & \frac{1}{8}-\frac{3\nu}{8} & \frac{\nu}{6} \\ -\frac{1}{4}-\frac{\nu}{12} & \frac{1}{8}-\frac{3\nu}{8} & \frac{1}{2}-\frac{\nu}{6} & -\frac{1}{8}-\frac{\nu}{8} \\ -\frac{1}{8}-\frac{3\nu}{8} & \frac{\nu}{6} & -\frac{1}{8}-\frac{\nu}{8} & \frac{1}{2}-\frac{\nu}{6} \end{pmatrix}$$

The strain displacement matrix $\mathbf{B}$ and the matrix for plane stress $\mathbf{E}$ are given as

$$\mathbf{B} = \begin{pmatrix} -0.5 & 0 & 0.5 & 0 & 0.5 & 0 & -0.5 & 0 \\ 0 & -0.5 & 0 & -0.5 & 0 & 0.5 & 0 & 0.5 \\ -0.5 & -0.5 & -0.5 & 0.5 & 0.5 & 0.5 & 0.5 & -0.5 \end{pmatrix}$$

and

$$\mathbf{E} = \frac{E}{(1-\nu^2)} \cdot \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{pmatrix} ,$$

respectively.

A.4 Correlation Coefficient

The correlation coefficient, or Pearson's correlation coefficient, is a measure for the linear correlation between to random variables X and Y. The

coefficient is defined as:

$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} \ ,$$

where $\mathrm{cov}(X,Y)$ is the covariance of X and Y. The coefficient r_{XY} is 1 for a total linear correlation, it is -1 for a total linear anti-correlation and 0 for no linear correlation. When dealing with two samples distributions for the variables we compute the empirical coefficients according to:

$$r = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}} \ ,$$

where x_i, y_i are the samples and $\bar{x}$, $\bar{y}$ are the sample means. More details can be found in standard maths handbooks.

B Compliance Studies Results

This section of the appendix collects result from the experiments on the minimum compliance problem in Chap. 5.

In order to retrace the variations of the resulting structures when using evolutionary tuning of update-signals, the resulting structures are shown. Figures B.1 and B.2 show the structures from running NE/PCM-TOPS with LSF Vector I. Figures B.3 and B.4 show the structures from running NE/PCM-TOPS with LSF Vector II. We see that NE-TOPS has a low variance for LSF Vector I and reproduces the reference structure reliably. For LSF Vector II variance in the structural designs increases, with a high proportion of structures with only two beams. Additional experiments that are not reported in this thesis, showed that the reference can be reproduced by reducing the move limit ($m = 0.02$) and increasing the allowed evaluations (e.g. $M = 2{,}000$, $M^{\max} = 10{,}000$) for model optimization. This requires significantly higher computational cost, compare also [14]. PCM-TOPS, however, is able to reproduce structures similar to the reference structure for the setup with LSF Vector I as well as for LSF Vector II.

The quantitative results of all experiments on the compliance problem are presented as well. Values considered are those for the compliance, i.e. its lowest $c_{\min}$, mean $\overline{c}$ and maximum value $c_{\max}$, as well as the total number of evaluations required by the optimization, i.e. the its lowest $M_{\text{tot,min}}$, mean $\overline{M}_{\text{tot}}$ and maximum $M_{\text{tot,max}}$ values. The standard deviations are indicated as well. Table B.1 shows the absolute values, Tab. B.2 shows the values relative to the according values of the finite differencing reference result, which is $c_{\text{ref}} = 54.45\text{Nmm}$ and $M_{\text{ref}} = 26{,}481$ for most cases with exception of the mesh-dependency study with SVR-TOPS. There, for simplicity, the same reference compliance value is used (since mesh dependency is expected), and the reference number of evaluations is estimated as $M_{\text{ref}} = (N_x \times N_y + 1) \times 21$, where N_x and N_y are the number of elements in horizontal and vertical directions, and 21 is the number of iterations required by the reference on the 45×28 mesh.

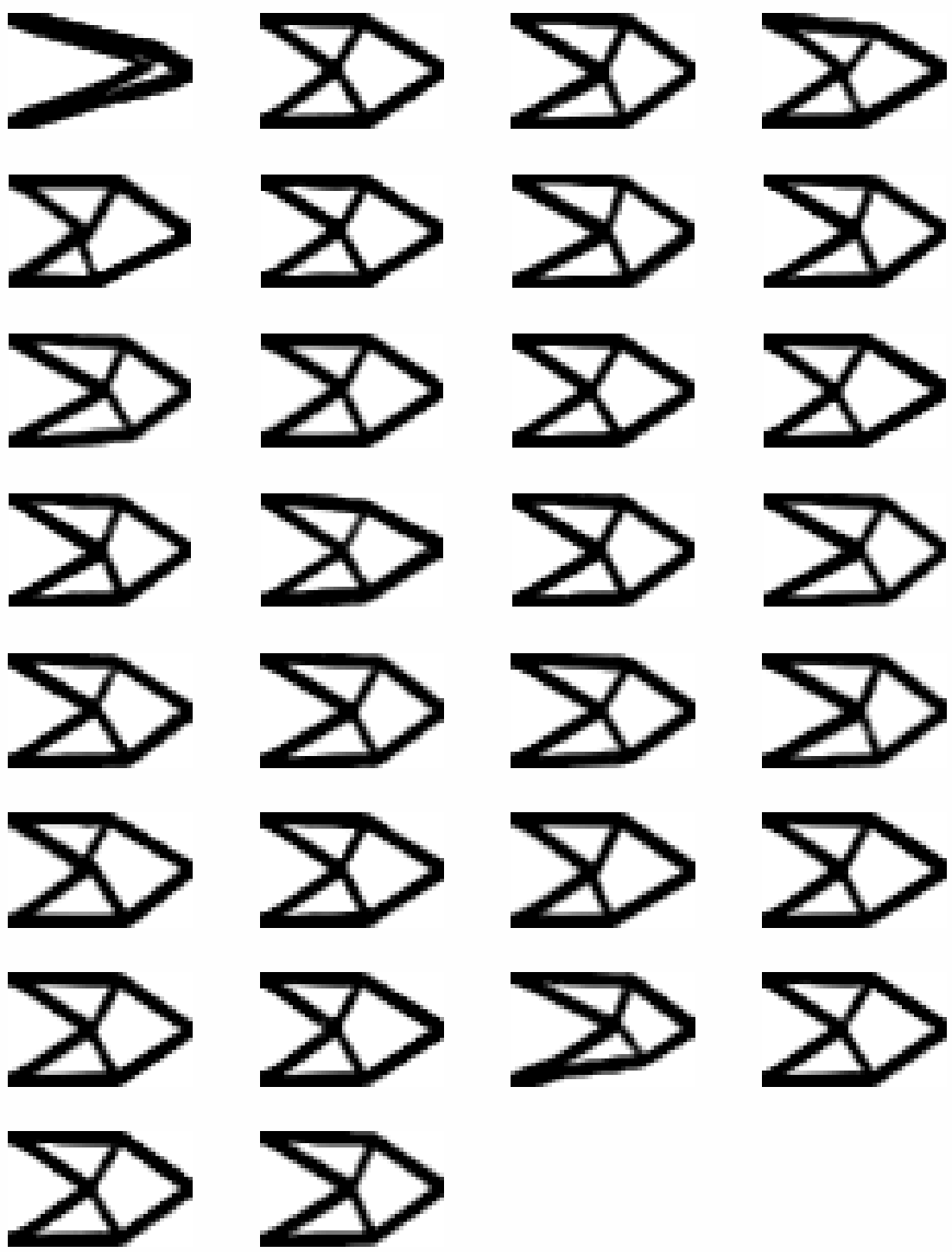


Figure B.1: The optimized structures resulting of running NE-TOPS with two hidden neurons for 30 different random seeds on the minimum compliance cantilever problem, for LSF Vector I. Structures very similar to the reference are obtained in the majority of cases.

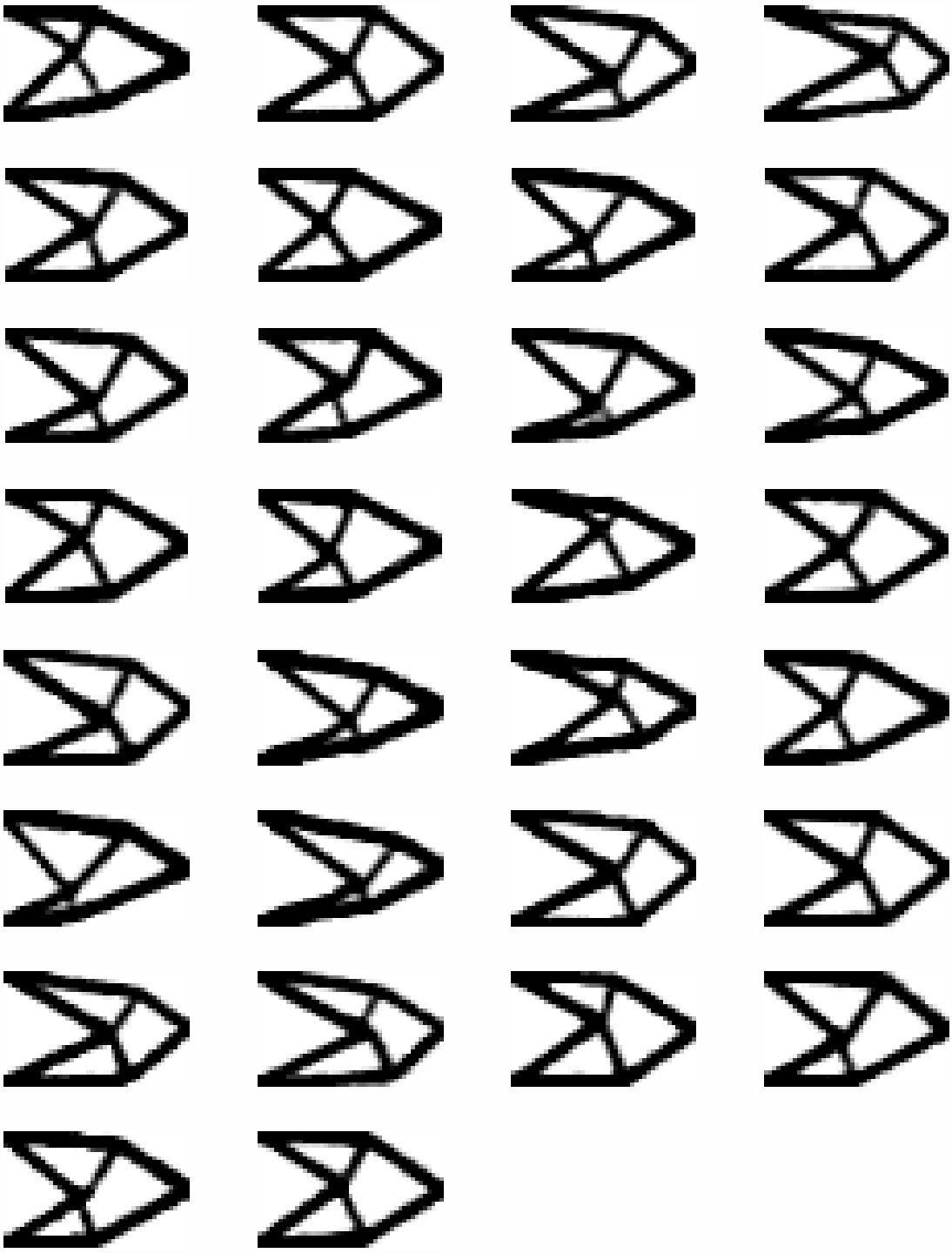


Figure B.2: The optimized structures resulting of running PCM-TOPS with 361 prototypes for 30 different random seeds on the minimum compliance cantilever problem, for LSF Vector I. Structures similar to the reference are obtained in the majority of cases.

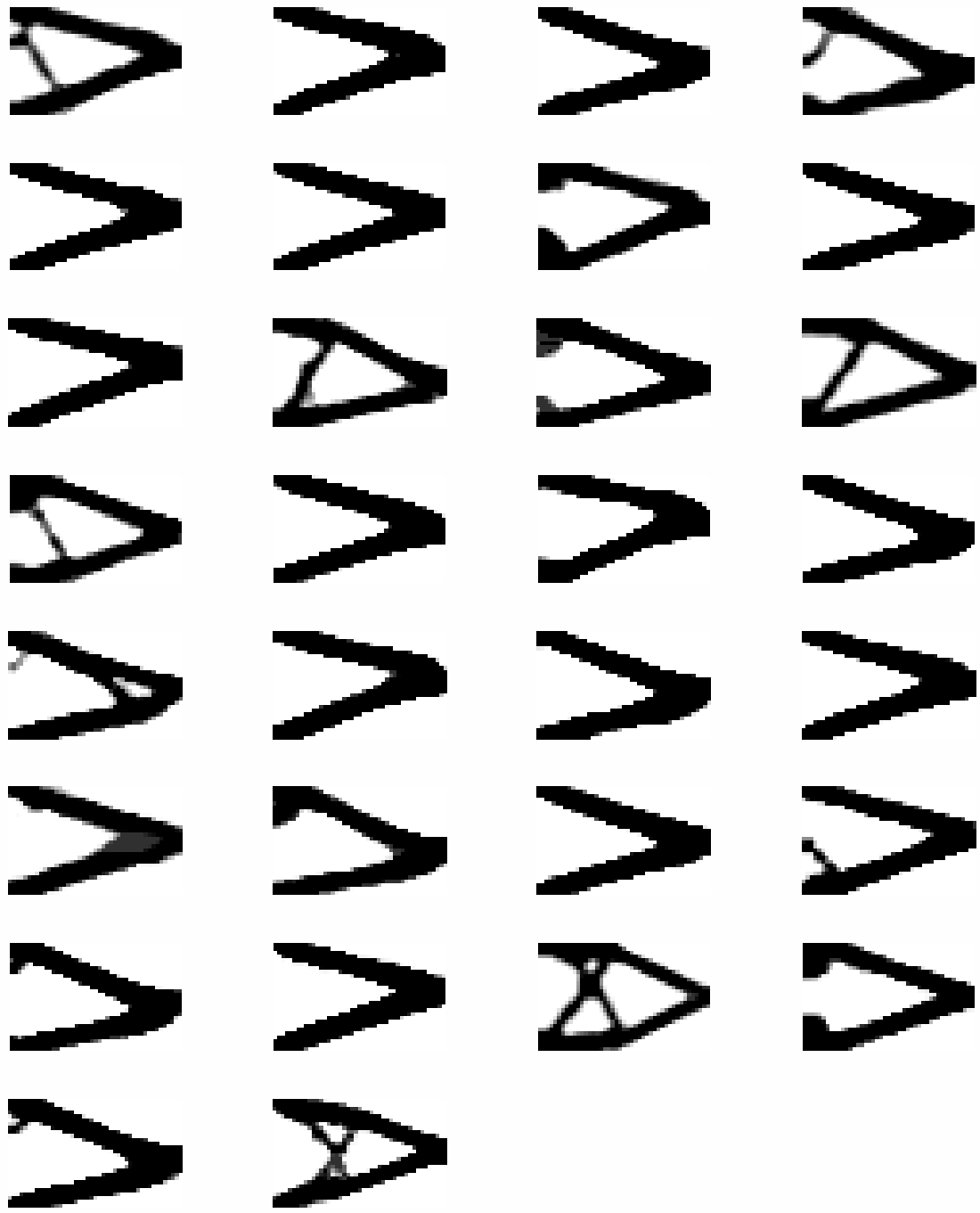


Figure B.3: The optimized structures resulting of running NE-TOPS with 30 hidden neurons for 30 different random seeds on the minimum compliance cantilever problem, for LSF Vector II. The MLP cannot be optimized sufficiently within the experiments to reproduce the reference structure.

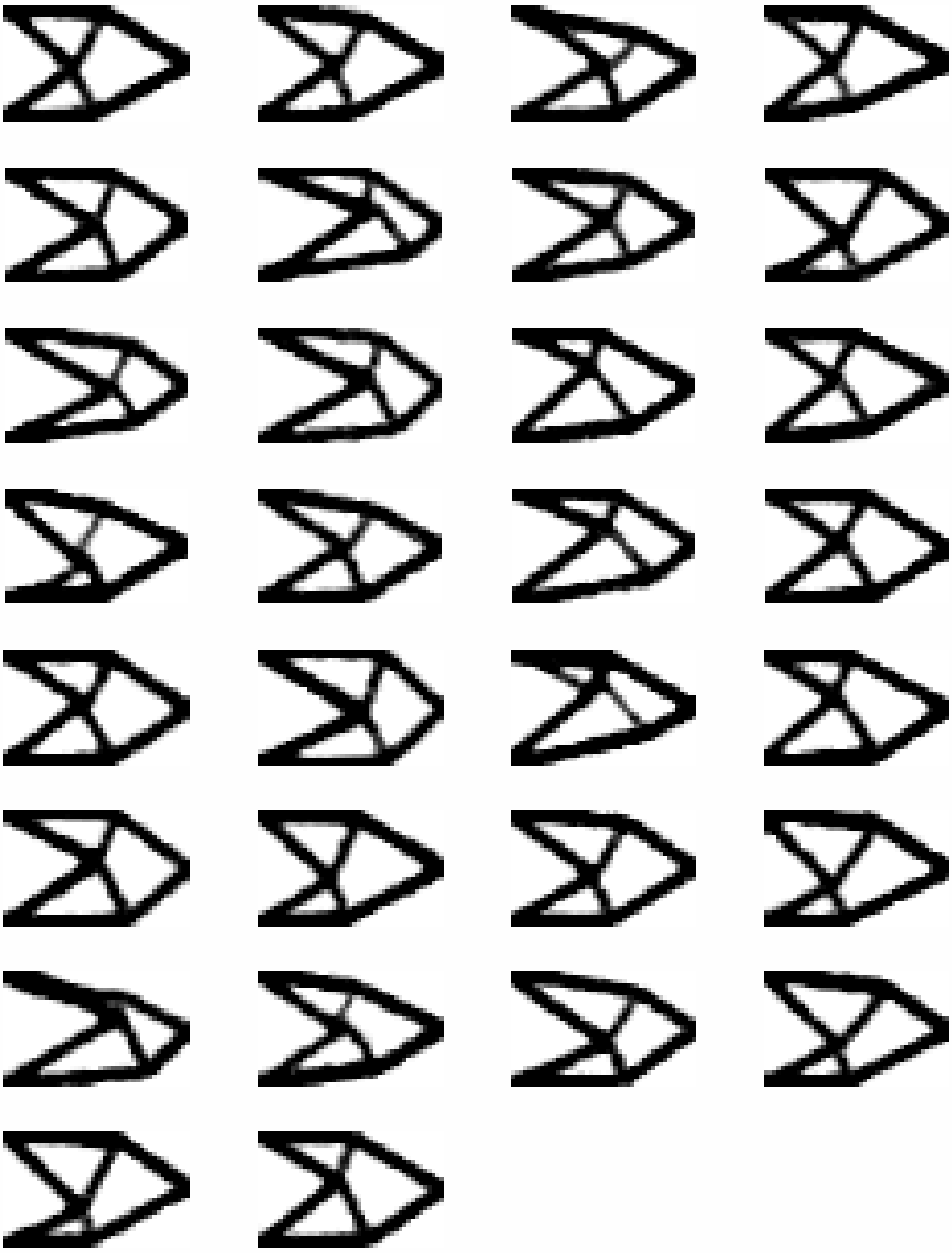


Figure B.4: The optimized structures resulting of running PCM-TOPS with 331 prototypes for 30 different random seeds on the minimum compliance cantilever problem, for LSF Vector II. Structures similar to the reference are obtained in the majority of cases.

Table B.1: Minimum, mean and maximum compliance and number of evaluations obtained by the different experiments on the compliance cantilever problem.

	$c_{\min}$/[Nmm]	$\overline{c}$ (±std) /[Nmm]	$c_{\max}$ /[Nmm]	$M_{\text{tot,min}}$	$\overline{M}_{\text{tot}}$ (±std)	$M_{\text{tot,max}}$
NE-TOPS						
$\mathbf{s}_i^{\text{I}}$, $H = 1$	51.68	52.64 (±1.59)	59.27	14042	17853.00 (±3659.93)	28080
$\mathbf{s}_i^{\text{I}}$, $H = 2$	51.18	52.36 (±0.99)	55.64	12048	18524.43 (±3959.67)	26070
$\mathbf{s}_i^{\text{I}}$, $H = 5$	51.42	53.22 (±1.43)	56.14	14029	21444.63 (±6205.33)	38067
$\mathbf{s}_i^{\text{I}}$, $H = 10$	52.8	54.89 (±0.87)	56.26	14121	20842.73 (±5170.86)	36306
$\mathbf{s}_i^{\text{I}}$, $H = 15$	51.91	54.81 (±0.99)	55.88	12091	24302.70 (±10202.50)	52351
$\mathbf{s}_i^{\text{I}}$, $H = 30$	53.74	55.10 (±0.63)	57.28	14119	24320.10 (±7605.26)	44303
$\mathbf{s}_i^{\text{II}}$, $H = 1$	149.78	226.61 (±34.25)	302.85	6012	13236.30 (±4691.16)	24327
$\mathbf{s}_i^{\text{II}}$, $H = 2$	61.17	137.20 (±55.48)	220.27	12027	25665.20 (±11990.52)	60149
$\mathbf{s}_i^{\text{II}}$, $H = 5$	53.97	58.28 (±4.15)	76.33	22034	39058.70 (±11327.13)	60094
$\mathbf{s}_i^{\text{II}}$, $H = 10$	53.95	56.35 (±2.53)	64.68	24210	42898.53 (±11296.45)	66569
$\mathbf{s}_i^{\text{II}}$, $H = 15$	53.59	56.08 (±2.28)	64.07	26201	45204.00 (±11172.13)	70477
$\mathbf{s}_i^{\text{II}}$, $H = 30$	53.62	55.97 (±2.12)	62.14	24210	43771.80 (±10525.18)	64495
PCM-TOPS						
$\mathbf{s}_i^{\text{I}}$, $P = 13$	57.45	63.65 (±4.46)	75.43	12031	20989.43 (±5310.63)	34090
$\mathbf{s}_i^{\text{I(red)}}$, $P = 13$	50.82	52.70 (±0.82)	54.77	6023	9968.57 (±2138.49)	16052
$\mathbf{s}_i^{\text{I}}$, $P = 25$	56.81	59.41 (±1.81)	64.93	12033	20987.43 (±3823.61)	28071
$\mathbf{s}_i^{\text{I}}$, $P = 61$	54.34	56.70 (±1.42)	61.65	14024	20364.03 (±3451.92)	28041
$\mathbf{s}_i^{\text{I}}$, $P = 121$	53.27	55.45 (±1.47)	59.93	16133	23855.50 (±6300.52)	38309
$\mathbf{s}_i^{\text{I}}$, $P = 181$	52.29	53.82 (±0.98)	55.73	18131	30320.07 (±8376.94)	52294
$\mathbf{s}_i^{\text{I}}$, $P = 361$	51.08	52.43 (±0.88)	54.57	20145	33625.43 (±6205.36)	48273
$\mathbf{s}_i^{\text{II}}$, $P = 12$	67.41	91.44 (±27.50)	203.24	12029	28070.87 (±6865.79)	40104
$\mathbf{s}_i^{\text{II}}$, $P = 23$	59.89	68.39 (±6.49)	82.25	20048	29206.13 (±6065.37)	40100
$\mathbf{s}_i^{\text{II}}$, $P = 56$	58.61	61.82 (±3.48)	75.01	16022	25769.57 (±6435.71)	44059
$\mathbf{s}_i^{\text{II}}$, $P = 111$	54.85	57.46 (±1.19)	60.04	20150	32453.40 (±9012.61)	54392
$\mathbf{s}_i^{\text{II}}$, $P = 166$	53.79	55.90 (±1.00)	59.17	18131	31741.13 (±6304.53)	44247
$\mathbf{s}_i^{\text{II}}$, $P = 331$	51.94	53.29 (±1.19)	56.34	26178	35708.40 (±7182.08)	62381
LIN-TOPS						
$\mathbf{s}_i^{\text{I}}$	53.07	53.57 (±0.33)	54.23	1550	1740.83 (±137.54)	2060
$\mathbf{s}_i^{\text{II}}$	58.79	61.20 (±2.22)	67.29	5117	11842.00 (±3983.47)	18975
$\mathbf{s}_i^{\text{II(2nd)}}$	56.86	59.07 (±2.16)	68.97	3196	6066.87 (±2124.02)	12584

$\mathbf{s}_i^{\text{II(3rd)}}$	54.66	55.94 (±0.89)	57.86	1133	2185.73 (±449.19)	3063
$\mathbf{s}_i^{\text{II(4th)}}$	54.84	56.05 (±0.96)	59.03	1033	1629.83 (±545.79)	3175
SVR-TOPS						
$\mathbf{s}_i^{\text{I}}$	53.76	54.37 (±0.35)	55.22	847	1530.10 (±431.02)	2479
$\mathbf{s}_i^{\text{II}}$	54.89	55.90 (±0.51)	56.89	4511	6022.27 (±930.85)	8067
$\mathbf{s}_i^{\text{III}}$	54.84	55.91 (±1.07)	60.09	906	3324.67 (±1377.25)	7367
$\mathbf{s}_i^{\text{IV}}$	55.07	56.58 (±1.54)	61.09	1566	2837.63 (±1094.27)	5431
$\mathbf{s}_i^{\text{V}}$	54.91	55.68 (±0.45)	56.36	4397	6305.50 (±1239.17)	9586
$\mathbf{s}_i^{\text{VI}}$	55.21	55.86 (±0.44)	57.07	4647	6629.63 (±1126.92)	8881
$\mathbf{s}_i^{\text{VII}}$	55.39	56.19 (±0.53)	58	1601	3350.53 (±941.52)	5527
$\mathbf{s}_i^{\text{VIII}}$	55.39	56.26 (±0.64)	58.37	1556	3126.60 (±1030.48)	6027
$\mathbf{s}_i^{\text{IX}}$	59.34	62.26 (±1.43)	65.82	5600	8842.30 (±1824.64)	13580
$\mathbf{s}_i^{\text{II}}$, 32×20	56.04	56.87 (±0.81)	59.44	4712	6031.83 (±960.67)	8753
$\mathbf{s}_i^{\text{II}}$, 45×28	55.58	56.77 (±0.96)	60.35	2583	5630.83 (±1129.04)	7366
$\mathbf{s}_i^{\text{II}}$, 64×40	56.63	63.18 (±13.56)	113.21	1541	4998.47 (±1825.36)	8265
$\mathbf{s}_i^{\text{II}}$, 96×60	58.48	79.88 (±27.49)	139.71	1527	3898.37 (±1820.37)	8349
$\mathbf{s}_i^{\text{II}}$, 128×80	57.97	74.73 (±27.95)	151.22	1628	4980.83 (±1971.94)	7946
SVR-TOPAS						
$N_{\mathcal{G}} = 1$, $\sigma_f = 0$	54.89	55.90 (±0.51)	56.89	4511	6022.27 (±930.85)	8067
$N_{\mathcal{G}} = 2$, $\sigma_f = 0$	55.02	56.28 (±0.79)	57.91	4087	6211.50 (±1163.38)	8664
$N_{\mathcal{G}} = 5$, $\sigma_f = 0$	55.38	58.00 (±6.81)	93.79	1453	5808.93 (±1572.19)	8985
$N_{\mathcal{G}} = 10$, $\sigma_f = 0$	56.35	58.31 (±5.11)	85.13	1752	5414.27 (±1287.75)	8290
$N_{\mathcal{G}} = 20$, $\sigma_f = 0$	57.76	59.04 (±1.53)	65.21	3600	5220.60 (±919.02)	7066
$N_{\mathcal{G}} = 30$, $\sigma_f = 0$	58.55	60.54 (±1.91)	68.8	2289	5202.73 (±871.28)	6558
$N_{\mathcal{G}} = 40$, $\sigma_f = 0$	60.24	63.03 (±3.90)	76.88	2503	5131.33 (±1156.00)	7759
$N_{\mathcal{G}} = 50$, $\sigma_f = 0$	60.72	66.47 (±7.78)	95.26	2087	4944.43 (±1187.89)	8476
$N_{\mathcal{G}} = 75$, $\sigma_f = 0$	63.53	84.74 (±22.35)	156.59	1532	3823.03 (±1414.40)	7051
$N_{\mathcal{G}} = 100$, $\sigma_f = 0$	79.55	147.55 (±53.50)	241.71	812	2900.37 (±1619.89)	7231
$N_{\mathcal{G}} = 1$, $\sigma_f = 0.005$	67.62	115.35 (±36.37)	192.49	931	3383.60 (±1986.76)	8244
$N_{\mathcal{G}} = 2$, $\sigma_f = 0.005$	60.67	75.16 (±19.47)	134.89	1445	5081.67 (±2148.98)	8954
$N_{\mathcal{G}} = 5$, $\sigma_f = 0.005$	57.46	59.43 (±1.73)	67.17	3214	5691.30 (±1332.73)	8359
$N_{\mathcal{G}} = 10$, $\sigma_f = 0.005$	56.77	59.47 (±6.59)	93.82	1444	5749.90 (±1315.12)	7962
$N_{\mathcal{G}} = 20$, $\sigma_f = 0.005$	57.83	59.35 (±0.96)	61.98	3892	5614.50 (±1186.25)	8164
$N_{\mathcal{G}} = 30$, $\sigma_f = 0.005$	58.81	60.84 (±2.15)	69.35	2912	5334.10 (±1015.07)	7674
$N_{\mathcal{G}} = 40$, $\sigma_f = 0.005$	60.62	63.54 (±3.27)	77.8	3003	5143.97 (±1041.52)	7781
$N_{\mathcal{G}} = 50$, $\sigma_f = 0.005$	61.62	66.58 (±4.83)	83.44	3072	5094.07 (±1047.77)	7459
$N_{\mathcal{G}} = 75$, $\sigma_f = 0.005$	65.34	79.92 (±17.65)	136.82	1533	4444.13 (±1582.62)	8179
$N_{\mathcal{G}} = 100$, $\sigma_f = 0.005$	72.54	143.63 (±53.95)	280.81	710	2657.57 (±1321.65)	5220

$N_{\mathcal{G}} = 1$, $\sigma_f = 0.01$	82.76	149.71 (±33.30)	212.77	812	2515.60 (±1385.03)	6317
$N_{\mathcal{G}} = 2$, $\sigma_f = 0.01$	73.64	108.08 (±33.36)	174.25	1034	3842.27 (±2206.99)	9136
$N_{\mathcal{G}} = 5$, $\sigma_f = 0.01$	59.8	69.34 (±13.72)	131.84	1541	5341.23 (±1745.53)	8673
$N_{\mathcal{G}} = 10$, $\sigma_f = 0.01$	58.32	61.42 (±4.75)	83.89	2186	5585.73 (±1453.56)	9902
$N_{\mathcal{G}} = 20$, $\sigma_f = 0.01$	57.87	60.78 (±1.41)	65.33	2789	5462.27 (±1248.46)	7766
$N_{\mathcal{G}} = 30$, $\sigma_f = 0.01$	58.9	61.71 (±3.08)	76.48	2709	5483.43 (±1261.01)	8167
$N_{\mathcal{G}} = 40$, $\sigma_f = 0.01$	61.14	63.53 (±1.77)	67.78	3713	5372.73 (±897.45)	7755
$N_{\mathcal{G}} = 50$, $\sigma_f = 0.01$	60.96	70.30 (±12.38)	111.89	2057	4732.70 (±1410.17)	7972
$N_{\mathcal{G}} = 75$, $\sigma_f = 0.01$	66.13	85.63 (±21.49)	146.71	1528	3890.87 (±1367.67)	7046
$N_{\mathcal{G}} = 100$, $\sigma_f = 0.01$	65.2	117.23 (±47.50)	240.93	813	3685.33 (±1659.10)	6643
$N_{\mathcal{G}} = 1$, $\sigma_f = 0.02$	134.3	249.22 (±71.06)	359.38	506	1392.70 (±1134.82)	5277
$N_{\mathcal{G}} = 2$, $\sigma_f = 0.02$	95.9	162.16 (±49.71)	273.96	638	2327.97 (±1407.98)	4878
$N_{\mathcal{G}} = 5$, $\sigma_f = 0.02$	71.75	115.91 (±31.83)	181.35	1119	2793.90 (±1347.12)	6011
$N_{\mathcal{G}} = 10$, $\sigma_f = 0.02$	62.44	74.87 (±12.98)	122.17	1326	4815.47 (±1531.84)	7744
$N_{\mathcal{G}} = 20$, $\sigma_f = 0.02$	59.86	63.36 (±2.09)	67.84	3628	5502.57 (±1067.74)	7457
$N_{\mathcal{G}} = 30$, $\sigma_f = 0.02$	61.17	64.39 (±3.22)	78.42	2174	5453.07 (±1123.84)	7873
$N_{\mathcal{G}} = 40$, $\sigma_f = 0.02$	59.96	64.72 (±2.72)	72.45	3409	5521.57 (±1039.07)	7066
$N_{\mathcal{G}} = 50$, $\sigma_f = 0.02$	63.55	72.08 (±9.91)	103.38	2277	4596.27 (±1485.69)	8674
$N_{\mathcal{G}} = 75$, $\sigma_f = 0.02$	64.42	86.62 (±20.56)	156.64	1424	3680.97 (±1305.01)	6948
$N_{\mathcal{G}} = 100$, $\sigma_f = 0.02$	71.75	126.60 (±42.84)	216.37	916	3135.73 (±1514.45)	6949

Table B.2: Minimum, mean and maximum compliance and number of evaluations obtained by the different experiments on the compliance cantilever problem relative to the reference.

	$c_{\min}/c_{\text{ref}}$	$\overline{c}/c_{\text{ref}}$ ($\pm$std)	$c_{\max}/c_{\text{ref}}$	$M_{\text{tot,min}}/M_{\text{ref}}$	$\overline{M}_{\text{tot}}/M_{\text{ref}}$ ($\pm$std)	$M_{\text{tot,max}}/M_{\text{ref}}$
NE-TOPS						
$\mathbf{s}_i^{\text{I}}$, $H = 1$	0.95	0.97 ($\pm$0.03)	1.09	0.53	0.67 ($\pm$0.14)	1.06
$\mathbf{s}_i^{\text{I}}$, $H = 2$	0.94	0.96 ($\pm$0.02)	1.02	0.45	0.70 ($\pm$0.15)	0.98
$\mathbf{s}_i^{\text{I}}$, $H = 5$	0.94	0.98 ($\pm$0.03)	1.03	0.53	0.81 ($\pm$0.23)	1.44
$\mathbf{s}_i^{\text{I}}$, $H = 10$	0.97	1.01 ($\pm$0.02)	1.03	0.53	0.79 ($\pm$0.20)	1.37
$\mathbf{s}_i^{\text{I}}$, $H = 15$	0.95	1.01 ($\pm$0.02)	1.03	0.46	0.92 ($\pm$0.39)	1.98
$\mathbf{s}_i^{\text{I}}$, $H = 30$	0.99	1.01 ($\pm$0.01)	1.05	0.53	0.92 ($\pm$0.29)	1.67
$\mathbf{s}_i^{\text{II}}$, $H = 1$	2.75	4.16 ($\pm$0.63)	5.56	0.23	0.50 ($\pm$0.18)	0.92
$\mathbf{s}_i^{\text{II}}$, $H = 2$	1.12	2.52 ($\pm$1.02)	4.05	0.45	0.97 ($\pm$0.45)	2.27
$\mathbf{s}_i^{\text{II}}$, $H = 5$	0.99	1.07 ($\pm$0.08)	1.4	0.83	1.47 ($\pm$0.43)	2.27
$\mathbf{s}_i^{\text{II}}$, $H = 10$	0.99	1.03 ($\pm$0.05)	1.19	0.91	1.62 ($\pm$0.43)	2.51
$\mathbf{s}_i^{\text{II}}$, $H = 15$	0.98	1.03 ($\pm$0.04)	1.18	0.99	1.71 ($\pm$0.42)	2.66
$\mathbf{s}_i^{\text{II}}$, $H = 30$	0.98	1.03 ($\pm$0.04)	1.14	0.91	1.65 ($\pm$0.40)	2.44
PCM-TOPS						
$\mathbf{s}_i^{\text{I}}$, $P = 13$	1.06	1.17 ($\pm$0.08)	1.39	0.45	0.79 ($\pm$0.20)	1.29
$\mathbf{s}_i^{\text{I(red)}}$, $P = 13$	0.93	0.97 ($\pm$0.02)	1.01	0.23	0.38 ($\pm$0.08)	0.61
$\mathbf{s}_i^{\text{I}}$, $P = 25$	1.04	1.09 ($\pm$0.03)	1.19	0.45	0.79 ($\pm$0.14)	1.06
$\mathbf{s}_i^{\text{I}}$, $P = 61$	1	1.04 ($\pm$0.03)	1.13	0.53	0.77 ($\pm$0.13)	1.06
$\mathbf{s}_i^{\text{I}}$, $P = 121$	0.98	1.02 ($\pm$0.03)	1.1	0.61	0.90 ($\pm$0.24)	1.45
$\mathbf{s}_i^{\text{I}}$, $P = 181$	0.96	0.99 ($\pm$0.02)	1.02	0.68	1.14 ($\pm$0.32)	1.97
$\mathbf{s}_i^{\text{I}}$, $P = 361$	0.94	0.96 ($\pm$0.02)	1	0.76	1.27 ($\pm$0.23)	1.82
$\mathbf{s}_i^{\text{II}}$, $P = 12$	1.24	1.68 ($\pm$0.50)	3.73	0.45	1.06 ($\pm$0.26)	1.51
$\mathbf{s}_i^{\text{II}}$, $P = 23$	1.1	1.26 ($\pm$0.12)	1.51	0.76	1.10 ($\pm$0.23)	1.51
$\mathbf{s}_i^{\text{II}}$, $P = 56$	1.08	1.14 ($\pm$0.06)	1.38	0.61	0.97 ($\pm$0.24)	1.66
$\mathbf{s}_i^{\text{II}}$, $P = 111$	1.01	1.06 ($\pm$0.02)	1.1	0.76	1.23 ($\pm$0.34)	2.05
$\mathbf{s}_i^{\text{II}}$, $P = 166$	0.99	1.03 ($\pm$0.02)	1.09	0.68	1.20 ($\pm$0.24)	1.67
$\mathbf{s}_i^{\text{II}}$, $P = 331$	0.95	0.98 ($\pm$0.02)	1.03	0.99	1.35 ($\pm$0.27)	2.36
LIN-TOPS						
$\mathbf{s}_i^{\text{I}}$	0.97	0.98 ($\pm$0.01)	1	0.06	0.07 ($\pm$0.01)	0.08
$\mathbf{s}_i^{\text{II}}$	1.08	1.12 ($\pm$0.04)	1.24	0.19	0.45 ($\pm$0.15)	0.72
$\mathbf{s}_i^{\text{II(2nd)}}$	1.04	1.08 ($\pm$0.04)	1.27	0.12	0.23 ($\pm$0.08)	0.48

$\mathbf{s}_i^{\text{II(3rd)}}$	1	1.03 (±0.02)	1.06	0.04	0.08 (±0.02)	0.12
$\mathbf{s}_i^{\text{II(4th)}}$	1.01	1.03 (±0.02)	1.08	0.04	0.06 (±0.02)	0.12
SVR-TOPS						
$\mathbf{s}_i^{\text{I}}$	0.99	1.00 (±0.01)	1.01	0.03	0.06 (±0.02)	0.09
$\mathbf{s}_i^{\text{II}}$	1.01	1.03 (±0.01)	1.04	0.17	0.23 (±0.04)	0.3
$\mathbf{s}_i^{\text{III}}$	1.01	1.03 (±0.02)	1.1	0.03	0.13 (±0.05)	0.28
$\mathbf{s}_i^{\text{IV}}$	1.01	1.04 (±0.03)	1.12	0.06	0.11 (±0.04)	0.21
$\mathbf{s}_i^{\text{V}}$	1.01	1.02 (±0.01)	1.04	0.17	0.24 (±0.05)	0.36
$\mathbf{s}_i^{\text{VI}}$	1.01	1.03 (±0.01)	1.05	0.18	0.25 (±0.04)	0.34
$\mathbf{s}_i^{\text{VII}}$	1.02	1.03 (±0.01)	1.07	0.06	0.13 (±0.04)	0.21
$\mathbf{s}_i^{\text{VIII}}$	1.02	1.03 (±0.01)	1.07	0.06	0.12 (±0.04)	0.23
$\mathbf{s}_i^{\text{IX}}$	1.09	1.14 (±0.03)	1.21	0.21	0.33 (±0.07)	0.51
$\mathbf{s}_i^{\text{II}}$, 32×20	1.03	1.04 (±0.01)	1.09	0.35	0.45 (±0.07)	0.65
$\mathbf{s}_i^{\text{II}}$, 45×28	1.02	1.04 (±0.02)	1.11	0.1	0.21 (±0.04)	0.28
$\mathbf{s}_i^{\text{II}}$, 64×40	1.04	1.16 (±0.25)	2.08	0.03	0.09 (±0.03)	0.15
$\mathbf{s}_i^{\text{II}}$, 96×60	1.07	1.47 (±0.50)	2.57	0.01	0.03 (±0.02)	0.07
$\mathbf{s}_i^{\text{II}}$, 128×80	1.06	1.37 (±0.51)	2.78	0.01	0.02 (±0.01)	0.04
SVR-TOPAS						
$N_{\mathcal{G}} = 1$, $\sigma_f = 0$	1.01	1.03 (±0.01)	1.04	0.17	0.23 (±0.04)	0.3
$N_{\mathcal{G}} = 2$, $\sigma_f = 0$	1.01	1.03 (±0.01)	1.06	0.15	0.23 (±0.04)	0.33
$N_{\mathcal{G}} = 5$, $\sigma_f = 0$	1.02	1.07 (±0.13)	1.72	0.05	0.22 (±0.06)	0.34
$N_{\mathcal{G}} = 10$, $\sigma_f = 0$	1.03	1.07 (±0.09)	1.56	0.07	0.20 (±0.05)	0.31
$N_{\mathcal{G}} = 20$, $\sigma_f = 0$	1.06	1.08 (±0.03)	1.2	0.14	0.20 (±0.03)	0.27
$N_{\mathcal{G}} = 30$, $\sigma_f = 0$	1.08	1.11 (±0.04)	1.26	0.09	0.20 (±0.03)	0.25
$N_{\mathcal{G}} = 40$, $\sigma_f = 0$	1.11	1.16 (±0.07)	1.41	0.09	0.19 (±0.04)	0.29
$N_{\mathcal{G}} = 50$, $\sigma_f = 0$	1.12	1.22 (±0.14)	1.75	0.08	0.19 (±0.04)	0.32
$N_{\mathcal{G}} = 75$, $\sigma_f = 0$	1.17	1.56 (±0.41)	2.88	0.06	0.14 (±0.05)	0.27
$N_{\mathcal{G}} = 100$, $\sigma_f = 0$	1.46	2.71 (±0.98)	4.44	0.03	0.11 (±0.06)	0.27
$N_{\mathcal{G}} = 1$, $\sigma_f = 0.005$	1.24	2.12 (±0.67)	3.54	0.04	0.13 (±0.08)	0.31
$N_{\mathcal{G}} = 2$, $\sigma_f = 0.005$	1.11	1.38 (±0.36)	2.48	0.05	0.19 (±0.08)	0.34
$N_{\mathcal{G}} = 5$, $\sigma_f = 0.005$	1.06	1.09 (±0.03)	1.23	0.12	0.21 (±0.05)	0.32
$N_{\mathcal{G}} = 10$ $\sigma_f = 0.005$	1.04	1.09 (±0.12)	1.72	0.05	0.22 (±0.05)	0.3
$N_{\mathcal{G}} = 20$, $\sigma_f = 0.005$	1.06	1.09 (±0.02)	1.14	0.15	0.21 (±0.04)	0.31
$N_{\mathcal{G}} = 30$, $\sigma_f = 0.005$	1.08	1.12 (±0.04)	1.27	0.11	0.20 (±0.04)	0.29
$N_{\mathcal{G}} = 40$, $\sigma_f = 0.005$	1.11	1.17 (±0.06)	1.43	0.11	0.19 (±0.04)	0.29
$N_{\mathcal{G}} = 50$, $\sigma_f = 0.005$	1.13	1.22 (±0.09)	1.53	0.12	0.19 (±0.04)	0.28
$N_{\mathcal{G}} = 75$, $\sigma_f = 0.005$	1.2	1.47 (±0.32)	2.51	0.06	0.17 (±0.06)	0.31
$N_{\mathcal{G}} = 100$, $\sigma_f = 0.005$	1.33	2.64 (±0.99)	5.16	0.03	0.10 (±0.05)	0.2

$N_{\mathcal{G}} = 1$, $\sigma_f = 0.01$	1.52	2.75 ($\pm$0.61)	3.91	0.03	0.09 ($\pm$0.05)	0.24
$N_{\mathcal{G}} = 2$, $\sigma_f = 0.01$	1.35	1.98 ($\pm$0.61)	3.2	0.04	0.15 ($\pm$0.08)	0.35
$N_{\mathcal{G}} = 5$, $\sigma_f = 0.01$	1.1	1.27 ($\pm$0.25)	2.42	0.06	0.20 ($\pm$0.07)	0.33
$N_{\mathcal{G}} = 10$, $\sigma_f = 0.01$	1.07	1.13 ($\pm$0.09)	1.54	0.08	0.21 ($\pm$0.05)	0.37
$N_{\mathcal{G}} = 20$, $\sigma_f = 0.01$	1.06	1.12 ($\pm$0.03)	1.2	0.11	0.21 ($\pm$0.05)	0.29
$N_{\mathcal{G}} = 30$, $\sigma_f = 0.01$	1.08	1.13 ($\pm$0.06)	1.4	0.1	0.21 ($\pm$0.05)	0.31
$N_{\mathcal{G}} = 40$, $\sigma_f = 0.01$	1.12	1.17 ($\pm$0.03)	1.24	0.14	0.20 ($\pm$0.03)	0.29
$N_{\mathcal{G}} = 50$, $\sigma_f = 0.01$	1.12	1.29 ($\pm$0.23)	2.05	0.08	0.18 ($\pm$0.05)	0.3
$N_{\mathcal{G}} = 75$, $\sigma_f = 0.01$	1.21	1.57 ($\pm$0.39)	2.69	0.06	0.15 ($\pm$0.05)	0.27
$N_{\mathcal{G}} = 100$, $\sigma_f = 0.01$	1.2	2.15 ($\pm$0.87)	4.42	0.03	0.14 ($\pm$0.06)	0.25
$N_{\mathcal{G}} = 1$, $\sigma_f = 0.02$	2.47	4.58 ($\pm$1.31)	6.6	0.02	0.05 ($\pm$0.04)	0.2
$N_{\mathcal{G}} = 2$, $\sigma_f = 0.02$	1.76	2.98 ($\pm$0.91)	5.03	0.02	0.09 ($\pm$0.05)	0.18
$N_{\mathcal{G}} = 5$, $\sigma_f = 0.02$	1.32	2.13 ($\pm$0.58)	3.33	0.04	0.11 ($\pm$0.05)	0.23
$N_{\mathcal{G}} = 10$, $\sigma_f = 0.02$	1.15	1.38 ($\pm$0.24)	2.24	0.05	0.18 ($\pm$0.06)	0.29
$N_{\mathcal{G}} = 20$, $\sigma_f = 0.02$	1.1	1.16 ($\pm$0.04)	1.25	0.14	0.21 ($\pm$0.04)	0.28
$N_{\mathcal{G}} = 30$, $\sigma_f = 0.02$	1.12	1.18 ($\pm$0.06)	1.44	0.08	0.21 ($\pm$0.04)	0.3
$N_{\mathcal{G}} = 40$, $\sigma_f = 0.02$	1.1	1.19 ($\pm$0.05)	1.33	0.13	0.21 ($\pm$0.04)	0.27
$N_{\mathcal{G}} = 50$, $\sigma_f = 0.02$	1.17	1.32 ($\pm$0.18)	1.9	0.09	0.17 ($\pm$0.06)	0.33
$N_{\mathcal{G}} = 75$, $\sigma_f = 0.02$	1.18	1.59 ($\pm$0.38)	2.88	0.05	0.14 ($\pm$0.05)	0.26
$N_{\mathcal{G}} = 100$, $\sigma_f = 0.02$	1.32	2.33 ($\pm$0.79)	3.97	0.03	0.12 ($\pm$0.06)	0.26

C Crashworthiness

This section contains:

- Details of the material model that was used in the crash simulations,
- plots of the update-signal models that resulted from the optimization of the crashworthiness objective functions.

In the crash experiments, a piecewise linear stress-strain relation is assumed. Figure C.1**(a)** shows the non-linear material model used for the beam crash case, for several values of density and penalization. Yield stress and tangent modulus are interpolated based on SIMP as proposed in [131]:

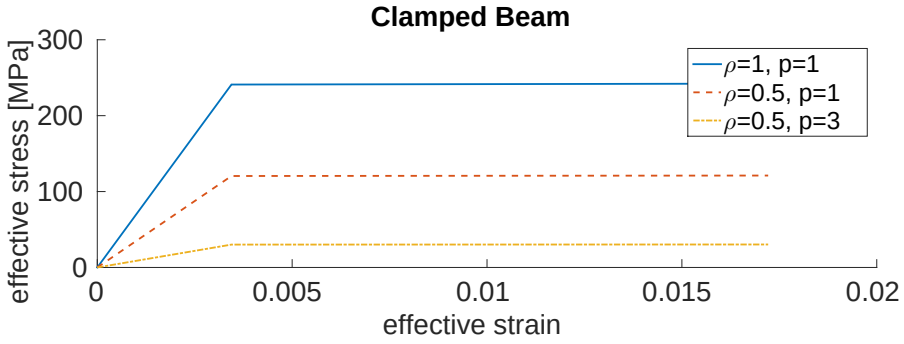
$$\begin{aligned} \varrho_i' &= \rho_i \varrho_0', \\ E_i &= \rho_i^p E_0, \\ \sigma_{\mathrm{y}i} &= \rho_i^q \sigma_{\mathrm{y}0}, \\ E_{\mathrm{h}i} &= \rho_i^p E_{\mathrm{h}0} \quad , \end{aligned} \tag{C.1}$$

with penalization constants p, q, volumetric mass density ϱ', yield stress σ_{y} and hardening modulus E_{h}.

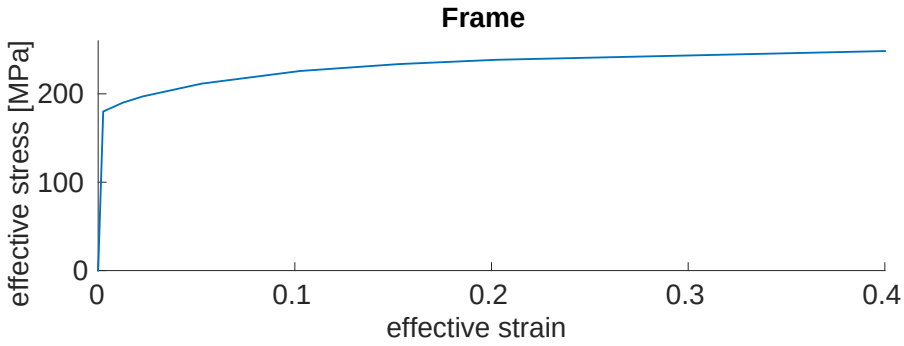
For the frame structure, a crash simulation with a piecewise linear model with more pieces is used, to account for the much higher deformations. It is shown in Fig. C.1**(b)**. The material parameters are shown in Tab. C.1. Since for the frame crash, the density variable represents a thickness, it is not necessary to penalize intermediate densities with SIMP as any value can be represented physically.

Figure C.2 shows the PCM update-signal model for the best run for some of the iterations of the energy maximization beam. Obviously, the optimized update-signal is not related linearly to the LSFs, a fact that justifies the usage of the non-linear PCM model. Also, there are visually significant changes between the iterations, showing that the re-optimization of the update-signal model leads to different update-signals. Often different update-signal values are possible for the same LSF value, indicating that

a second dimension of the LSF vector is necessary as well. Similar observations can be made as well for the mean run of the intrusion frame minimization shown in Fig. C.3.



(a) Piecewise linear material model used for elastic-plastic deformations of aluminum material model used in the beam crash case, shown for different elemental densities and penalizations.



(b) The strain-stress curve of the piecewise linear material model used for elastic-plastic deformations of the aluminum frame model based on the parameters in Tab. C.1.

Figure C.1: Stress-strain curves for the non-linear material models used in the crash simulations.

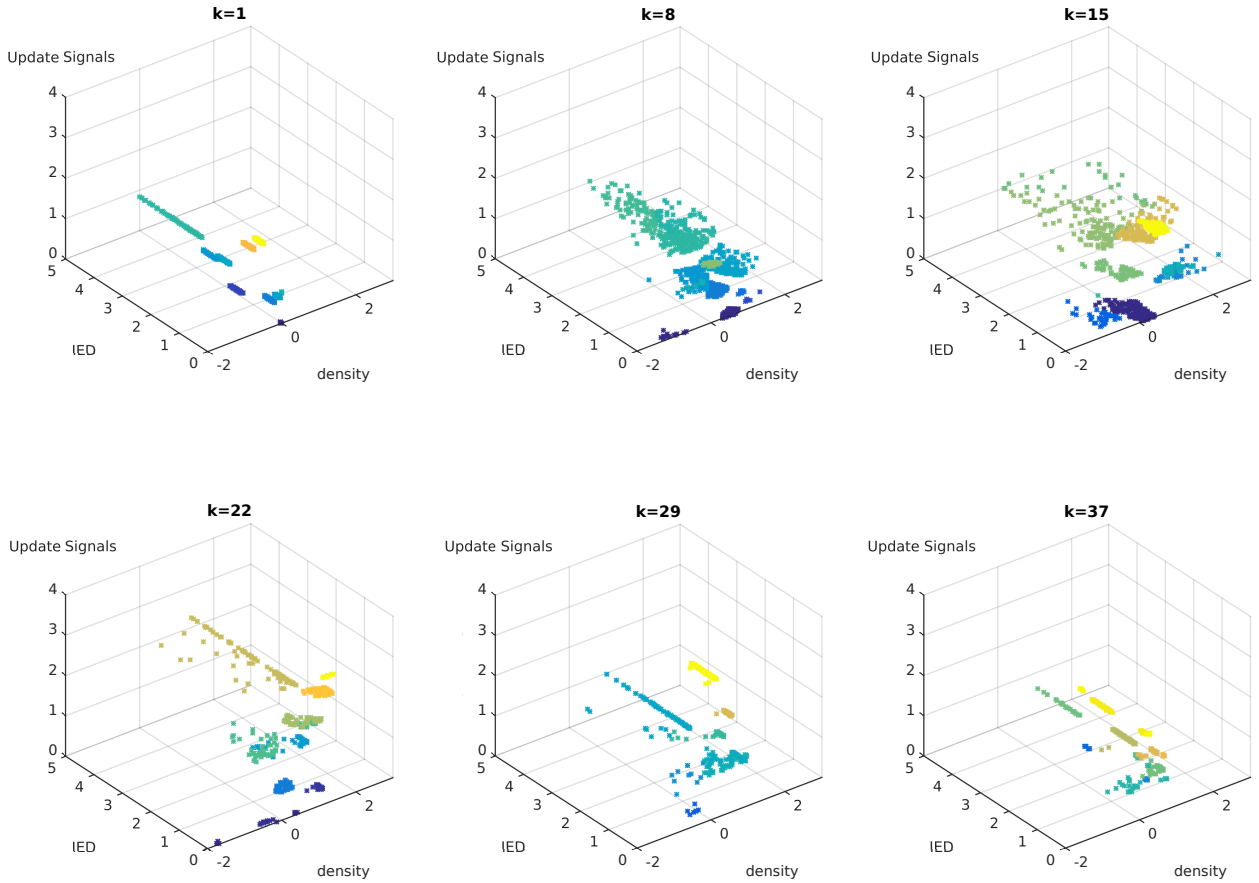


Figure C.2: The optimized update signals depending on the LSFs for the energy absorption beam for selected iterations.

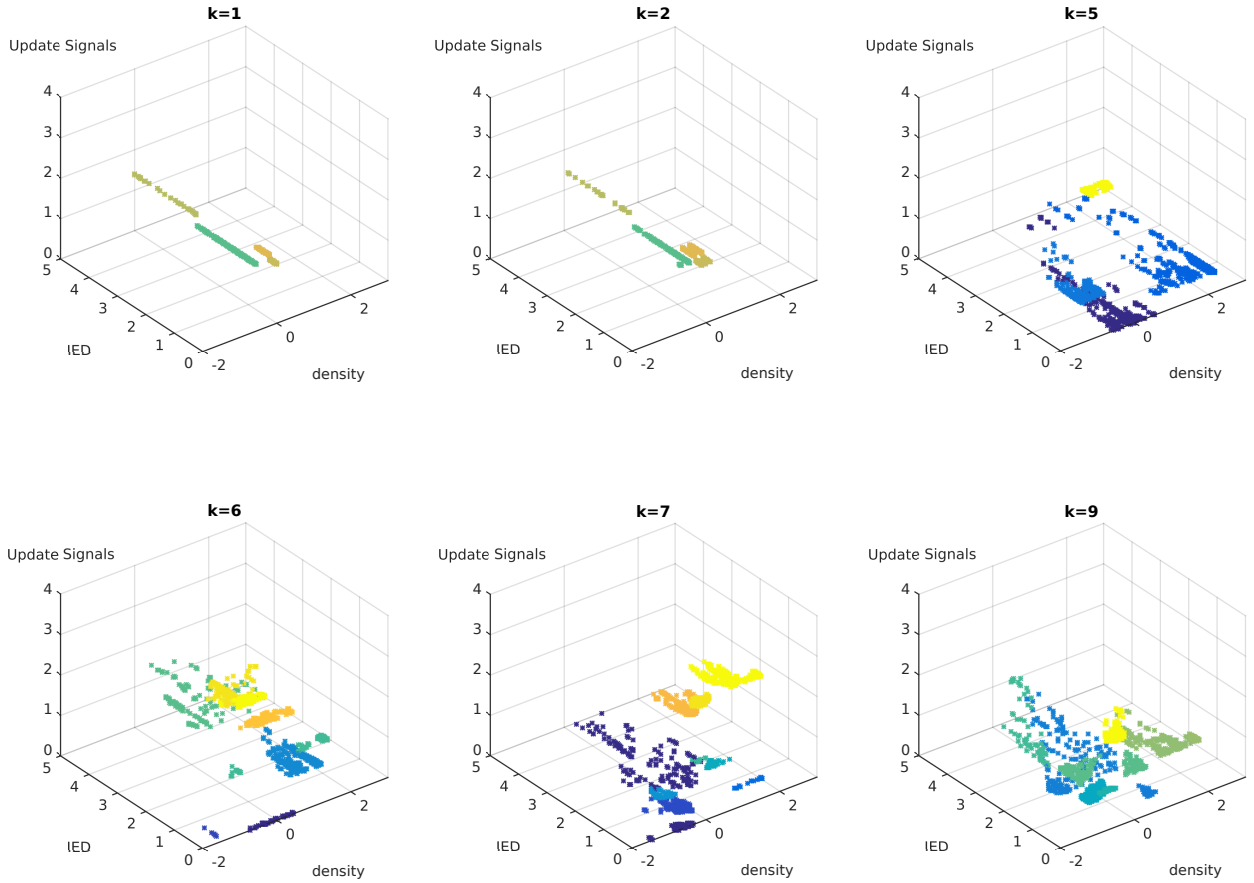


Figure C.3: The optimized update signals depending on the LSFs for the minimum intrusion frame.

Table C.1: Material constants of the elastic-plastic material model of the frame problem in Sec. 6.3.

Effective plastic strain	Effective yield stress [MPa]
0.01	190.0
0.02	197.0
0.05	211.5
0.10	225.8
0.15	233.6
0.20	238.5
0.40	248.5

Bibliography

[1] Adelman, H. M. and Haftka, R. T. Sensitivity analysis of discrete structural systems. *AIAA journal*, 24(5):823–832, 1986.

[2] Ahmed, F., Bhattacharya, B., and Deb, K. Constructive solid geometry based topology optimization using evolutionary algorithm. In Bansal, J. C., Singh, P. K., Deep, K., Pant, M., and Nagar, A. K., editors, *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012)*, volume 201 of *Advances in Intelligent Systems and Computing*, pages 227–238. Springer India, 2013.

[3] Allaire, G. and Jouve, F. Coupling the level set method and the topological gradient in structural optimization. In *IUTAM symposium on topological design optimization of structures, machines and materials*, pages 3–12. Springer, 2006.

[4] Allaire, G., Jouve, F., and Toader, A.-M. Structural optimization using sensitivity analysis and a level-set method. *Journal of computational physics*, 194(1):363–393, 2004.

[5] Andreassen, E., Clausen, A., Schevenels, M., Lazarov, B. S., and Sigmund, O. Efficient topology optimization in MATLAB using 88 lines of code. *Structural and Multidisciplinary Optimization*, 43(1): 1–16, 2011.

[6] Arora, J. S. and Haug, E. J. Methods of design sensitivity analysis in structural optimization. *AIAA journal*, 17(9):970–974, 1979.

[7] Auger, A. and Hansen, N. A restart CMA evolution strategy with increasing population size. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 2, pages 1769–1776. IEEE, 2005.

[8] Auger, A. and Hansen, N. Performance evaluation of an advanced local search evolutionary algorithm. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 2, pages 1777–1784. IEEE, 2005.

[9] Aulig, N. and Olhofer, M. Topology optimization by predicting sensitivities based on local state features. In *5th. European Conference on Computational Mechanics (ECCM V)*, Barcelona, Spain, 2014.

[10] Aulig, N. and Olhofer, M. State-based representation for structural topology optimization and application to crashworthiness. In *Evolutionary Computation (CEC), 2016 IEEE Congress on*, pages 1642–1649, Vancouver, BC, Canada, July 2016.

[11] Aulig, N. and Olhofer, M. Evolutionary computation for topology optimization of mechanical structures: An overview of representations. In *Evolutionary Computation (CEC), 2016 IEEE Congress on*, pages 1948–1955, Vancouver, BC, Canada, July 2016.

[12] Aulig, N. Evolutionary optimization of the robustness and evolvability of design solutions. Diploma thesis, Technische Universität Darmstadt, 2011.

[13] Aulig, N. and Lepenies, I. A topology optimization interface for LS-DYNA. In *11. LS-DYNA Forum*, Ulm, 2012.

[14] Aulig, N. and Olhofer, M. Neuro-evolutionary topology optimization of structures by utilizing local state features. In *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation*, GECCO '14, pages 967–974, New York, NY, USA, 2014. ACM.

[15] Aulig, N. and Olhofer, M. Neuro-evolutionary topology optimization with adaptive improvement threshold. In Mora, A. M. and Squillero, G., editors, *Applications of Evolutionary Computation*, volume 9028 of *Lecture Notes in Computer Science*, pages 655–666. Springer International Publishing, 2015.

[16] Aulig, N., Menzel, S., Nutwell, E., and Detwiler, D. Towards multi-objective topology optimization of structures subject to crash and static load cases. In *Engineering Optimization 2014*, pages 847–852. CRC Press, September 2014.

[17] Aulig, N., Menzel, S., Nutwell, E., and Detwiler, D. A weight balanced multi-objective topology optimization for automotive development. In *10th European LS-DYNA Conference 2015*, Würzburg, Germany, 2015.

[18] Aulig, N., Menzel, S., Nutwell, E., and Detwiler, D. Preference-based topology optimization of body-in-white structures for crash and static loads. In *14th LS-DYNA International Conference*, Dearborn, MI, USA, June 2016.

[19] Bäck, T., Fogel, D. B., and Michalewicz, T. *Evolutionary Computation 1: Basic Algorithms and Operators*. IOP Publishing Ltd, 2000.

[20] Bäck, T. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, Oxford, UK, 1996.

[21] Bäck, T., Foussette, C., and Krause, P. *Contemporarey Evolution Strategies*. 1619-7127. Springer Verlagg Berlinn Heidelberg, 2013.

[22] Balamurugan, R., Ramakrishnan, C., and Singh, N. Performance evaluation of a two stage adaptive genetic algorithm (TSAGA) in structural topology optimization. *Applied Soft Computing*, 8(4):1607 – 1624, 2008. Soft Computing for Dynamic Data Mining.

[23] Balamurugan, R., Ramakrishnan, C., and Swaminathan, N. A two phase approach based on skeleton convergence and geometric variables for topology optimization using genetic algorithm. *Structural and Multidisciplinary Optimization*, 43(3):381–404, 2011.

[24] Bandi, P. *Design of crashworthy structures with controlled behavior in HCA framework*. PhD thesis, University of Notre Dame, 2012.

[25] Baumgartner, A., Harzheim, L., and Mattheck, C. SKO (soft kill option): the biological way to find an optimum structure topology. *International Journal of Fatigue*, 14(6):387 – 393, 1992.

[26] Bendsøe, M. P. *Optimization of structural topology, shape, and material*. Springer-Verlag Berlin Heidelberg, 1995.

[27] Bendsøe, M. P. and Sigmund, O. Material interpolation schemes in topology optimization. *Archive of applied mechanics*, 69(9-10): 635–654, 1999.

[28] Bendsøe, M. Optimal shape design as a material distribution problem. *Structural optimization*, 1:193–202, 1989.

[29] Bendsøe, M. and Kikuchi, N. Generating optimal topologies in optimal design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71:197–224, 1988.

[30] Bendsøe, M. and Sigmund, O. *Topology Optimization Theory, Methods and Applications*. Springer Verlag Berlin, 2nd edition, 2004.

[31] Beyer, H.-G. and Schwefel, H.-P. Evolution strategies – a comprehensive introduction. *Natural Computing*, 1(1):3–52, 2002.

[32] Bishop, C. M. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[33] Brenn, M. Dreidimensionale topologieoptimierung mit hilfe eines zellwachstummodells. Master's thesis, Technische Universität Darmstadt, 2008.

[34] Bujny, M., Aulig, N., Olhofer, M., and Duddeck, F. Hybrid evolutionary approach for level set topology optimization. In *Evolutionary Computation (CEC), 2016 IEEE Congress on*, pages 5092–5099, Vancouver, BC, Canada, July 2016.

[35] Bujny, M. Development of a hybrid evolutionary approach for level set topology optimization. Master's thesis, Technische Universität München, 2015.

[36] Burger, M., Hackl, B., and Ring, W. Incorporating topological derivatives into level set methods. *Journal of Computational Physics*, 194(1):344–362, 2004.

[37] Campelo, F., Watanabe, K., and Igarashi, H. 3d topology optimization using an immune algorithm. *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, 26(3):677–688, 2007.

[38] Cavazzuti, M., Baldini, A., Bertocchi, E., Costi, D., Torricelli, E., and Moruzzi, P. High performance automotive chassis design: a topology optimization based approach. *Structural and Multidisciplinary Optimization*, 44(1):45–56, 2011.

[39] Chan, C.-M. and Wong, K.-M. Structural topology and element sizing design optimisation of tall steel frameworks using a hybrid OC-GA method. *Structural and Multidisciplinary Optimization*, 35 (5):473–488, 2008.

[40] Chandra, A. and Yao, X. Ensemble learning using multi-objective evolutionary algorithms. *J. Math. Model. Algorithms*, 5(4):417–445, 2006.

[41] Chang, C.-C. and Lin, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[42] Chapman, C. D., Saitou, K., and Jakiela, M. J. Genetic algorithms as an approach to configuration and topology design. *Journal of Mechanical Design*, 116(4):1005–1012, 1994.

[43] Chen, P., Chen, C., Wang, H., Tsai, J., and Ni, W.-X. Synthesis design of artificial magnetic metamaterials using a genetic algorithm. *Optics express*, 16(17):12806–12818, 2008.

[44] Chen, Z., Gao, L., Qiu, H., and Shao, X. A GAOC method for topology optimization design. In *Intelligent Computation Technology and Automation, 2009. ICICTA '09. Second International Conference on*, volume 3, pages 293–296, Oct 2009.

[45] Cheney, N., Ritz, E., and Lipson, H. Automated vibrational design and natural frequency tuning of multi-material structures. In *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation*, GECCO '14, pages 1079–1086, New York, NY, USA, 2014. ACM.

[46] Christensen, J., Bastien, C., and Blundell, M. Effects of roof crush loading scenario upon body in white using topology optimisation. *International Journal of Crashworthiness*, 17(1):29–38, 2012.

[47] Darwin, C. *The Origin of Species by Means of Natural Selection.* London: Murray, 1859.

[48] Dawkins, R. *The selfish gene.* Oxford university press, 1976.

[49] Dawkins, R. *The blind watchmaker.* Norton, New York, 1986.

[50] Dawkins, R. The evolution of evolvability. In Kumar, S., editor, *On Growth, Form and Computers*, pages 239 – 255, San Diego, CA, 2003. Elsevier.

[51] De Ruiter, M. and Van Keulen, F. Topology optimization using a topology description function. *Structural and Multidisciplinary Optimization*, 26(6):406–416, 2004.

[52] Deaton, J. D. and Grandhi, R. V. A survey of structural and multidisciplinary continuum topology optimization: post 2000. *Structural and Multidisciplinary Optimization*, 49(1):1–38, 2014.

[53] Denies, J. *Métaheuristiques pour l'optimisation topologique: application à la conception de dispositifs électromagnétiques*. PhD thesis, École normale supérieure de Cachan-ENS Cachan, 2013.

[54] Denies, J., Dehez, B., Glineur, F., and Ben Ahmed, H. Genetic algorithm-based topology optimization: Performance improvement through dynamic evolution of the population size. In *Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM), 2012 International Symposium on*, pages 1033–1038. IEEE, 2012.

[55] Dennis, J. E. and Schnabel, R. B. A view of unconstrained optimization. *Optimization*, pages 1–72, Elsevier, New York 1987.

[56] Diaz, A. and Sigmund, O. Checkerboard patterns in layout optimization. *Structural optimization*, 10(1):40–45, 1995.

[57] Duddeck, F. Multidisciplinary optimization of car bodies. *Structural and Multidisciplinary Optimization*, 35(4):375–389, 2008.

[58] Duddeck, F., Hunkeler, S., Lozano, P., Wehrle, E., and Zeng, D. Topology optimization for crashworthiness of thin-walled structures under axial impact using hybrid cellular automata. *Structural and Multidisciplinary Optimization*, pages 1–14, 2016.

[59] Eiben, A. and Smith, J. *Introduction to Evolutionary Computation*. Natural Computing Series. Springer Berlin Heidelberg, 2nd edition edition, 2007.

[60] Eschenauer, H., Kobelev, V., and Schumacher, A. Bubble method for topology and shape optimization of structures. *Structural optimization*, 8(1):42–51, 1994.

[61] European New Car Assessment Programme (Euro NCAP). Oblique pole side impact testing protocol version 7.0.2, November 2015.

[62] Evins, R., Vaidyanathan, R., and Burges, S. Multi-material compositional pattern-producing networks for form optimisation. In *European Conference on the Applications of Evolutionary Computation, EvoApplications*, 2014.

[63] Floreano, D., Dürr, P., and Mattiussi, C. Neuroevolution: from architectures to learning. *Evolutionary Intelligence*, 1(1):47–62, 2008.

[64] Fogel, L. J., Owens, A. J., and Walsh, M. J. *Artificial Intelligence through Simulated Evolution*. John Wiley, New York, USA, 1966.

[65] Forrester, A., Sobester, A., and Keane, A. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.

[66] Forsberg, J. and Nilsson, L. Topology optimization in crashworthiness design. *Structural and Multidisciplinary Optimization*, 33:1–12, 2007. 10.1007/s00158-006-0040-z.

[67] Fredricson, H. Structural topology optimisation: an application review. *International journal of vehicle design*, 37(1):67–80, 2005.

[68] Giles, M. B. and Pierce, N. A. An introduction to the adjoint approach to design. *Flow, turbulence and combustion*, 65(3-4):393–415, 2000.

[69] Goetz, J., Tan, H., Renaud, J., and Tovar, A. Two-material optimization of plate armour for blast mitigation using hybrid cellular automata. *Engineering Optimization*, 44(8):985–1005, 2012.

[70] Gomez, F., Schmidhuber, J., and Miikkulainen, R. Accelerated neural evolution through cooperatively coevolved synapses. *The Journal of Machine Learning Research*, 9:937–965, 2008.

[71] Guirguis, D., Hamza, K., Aly, M., Hegazi, H., and Saitou, K. Multi-objective topology optimization of multi-component continuum structures via a kriging-interpolated level set approach. *Structural and Multidisciplinary Optimization*, 51(3):733–748, 2015.

[72] Hamda, H. and Schoenauer, M. Topological optimum design with evolutionary algorithms. *Journal of Convex Analysis*, 9(2):503–518, 2002.

[73] Hamda, H., Jouve, F., Lutton, E., Schoenauer, M., and Sebag, M. Compact unstructured representations for evolutionary design. *Applied Intelligence*, 16(2):139–155, 2002.

[74] Hamza, K., Aly, M., and Hegazi, H. An explicit level-set approach for structural topology optimization. In *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2013.

[75] Han, Y. H., Witowski, K., Lazatov, N., and Anakiev, K. Topometry and shape optimization of a hood. In *10th European LS-DYNA Conference 2015*, Würzburg, Germany, 2015.

[76] Hansen, N. The CMA evolution strategy: A comparing review. In Lozano, J., Larrañaga, P., Inza, I. n., and Bengoetxea, E., editors, *Towards a New Evolutionary Computation*, volume 192 of *Studies in Fuzziness and Soft Computing*, pages 75–102. Springer Berlin Heidelberg, 2006.

[77] Hansen, N. and Kern, S. Evaluating the CMA evolution strategy on multimodal test functions. In *Parallel Problem Solving from Nature, PPSN 2004*, 2004.

[78] Hansen, N. and Ostermeier, A. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2): 159–195, 2001.

[79] Hansen, N., Auger, A., Ros, R., Finck, S., and Pošík, P. Comparing results of 31 algorithms from the black-box optimization benchmarking bbob-2009. In *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*, pages 1689–1696. ACM, 2010.

[80] Hauschild, M. and Pelikan, M. An introduction and survey of estimation of distribution algorithms. *Swarm and Evolutionary Computation*, 1(3):111–128, 2011.

[81] Heidrich-Meisner, V. and Igel, C. Similarities and differences between policy gradient methods and evolution strategies. In *ESANN*, pages 149–154. Citeseer, 2008.

[82] Heidrich-Meisner, V. and Igel, C. Neuroevolution strategies for episodic reinforcement learning. *J. Algorithms*, 64(4):152–168, 2009.

[83] Hoerl, A. E. and Kennard, R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

[84] Holland, J. H. Outline for a logical theory of adaptive systems. *J. ACM*, 9:297–314, July 1962.

[85] Hornby, G. S. Creating complex building blocks through generative representations. *Proceedings of the 2003 AAAI Spring Symposioum: Computational Synthesis: From Basic Building Blocks to High Level Functionality, American Association for Artificial Intelligence, AAAI Press*, pages 98–105, 2003.

[86] Hornby, G. S. Generative representations for evolving families of designs. In *Genetic and Evolutionary Computation - GECCO 2003*, pages 1678–1689. Springer, 2003.

[87] Hsu, C.-W., Chang, C.-C., and Lin, C.-J. A practical guide to support vector classification. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, Taiwan, 2003.

[88] Hu, T. and Banzhaf, W. Evolvability and speed of evolutionary algorithms in light of recent developments in biology. *Journal of Artificial Evolution and Applications*, 2010.

[89] Huang, X. and Xie, Y. *Evolutionary Topology Optimization of continuum structures.* John Wiley & Sons, 2012.

[90] Huang, X., Xie, Y. M., and Lu, G. Topology optimization of energy-absorbing structures. *International Journal of Crashworthiness*, 12 (6):663–675, 2007.

[91] Huang, X. and Xie, M. *Evolutionary topology optimization of continuum structures: methods and applications.* John Wiley & Sons, 2010.

[92] Hunkeler, S. *Topology Optimisation in Crashworthiness Design via Hybrid Cellular Automata for Thin-walled Structures.* PhD thesis, Queen Mary University of London, UK, 2013.

[93] Igel, C. Neuroevolution for reinforcement learning using evolution strategies. In *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*, volume 4, pages 2588 – 2595, dec. 2003.

[94] Jain, C. and Saxena, A. An improved material-mask overlay strategy for topology optimization of structures and compliant mechanisms. *Journal of Mechanical Design*, 132(6):061006, 2010.

[95] Jastrebski, G. A. and Arnold, D. V. Improving evolution strategies through active covariance matrix adaptation. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 2814–2821, 2006.

[96] Jin, Y. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61–70, 2011.

[97] Jo, H.-C., Yoo, K.-S., Park, J.-Y., and Han, S.-Y. Dynamic topology optimization based on ant colony optimization. In *Natural Computation (ICNC), 2012 Eighth International Conference on*, pages 763–766, May 2012.

[98] Kang, B., Choi, W., and Park, G. Structural optimization under equivalent static loads transformed from dynamic loads based on displacement. *Computers & Structures*, 79(2):145–154, 2001.

[99] Karush, W. Minima of functions of several variables with inequalities as side constraints. Master's thesis, Dept. of Mathematics, Univ. of Chicago, 1939.

[100] Kaveh, A., Hassani, B., Shojaee, S., and Tavakkoli, S. Structural topology optimization using ant colony methodology. *Engineering Structures*, 30(9):2559 – 2565, 2008.

[101] Kicinger, R., Arciszewski, T., and De Jong, K. Evolutionary computation and structural design: A survey of the state-of-the-art. *Computers & Structures*, 83(23):1943–1978, 2005.

[102] Kim, S.-C., Chang, D.-H., Yoo, K.-S., and Han, S.-Y. Development of modified ant colony optimization algorithm for compliant mechanisms. In *SIMUL 2012, The Fourth International Conference on Advances in System Simulation*, pages 152–157, 2012.

[103] Kim, Y. and De Weck, O. Progressive structural topology optimization by variable chromosome length genetic algorithm. In *China-Japan-Korea Joint Symposium on Optimization of Structural and Mechanical Systems, M3*, 2004.

[104] Kobayashi, M. H., Pedro, H.-T. C., Kolonay, R. M., and Reich, G. W. On a cellular division method for aircraft structural design. *The Aeronautical Journal*, 113:821–831, 2009.

[105] Kuhn, H. W. and Tucker, A. W. Nonlinear programming. In *Neyman: Proceedings of the 2nd Berkely Symp. on Math. Statistics and Probability*, pages 481–492. Berkeley: University of California Press, 1951.

[106] Larranaga, P. and Lozano, J. A. *Estimation of distribution algorithms: A new tool for evolutionary computation*, volume 2. Springer Science & Business Media, 2002.

[107] Li, C., Hiroyasu, T., and Miki, M. Stress-based crossover operator for structural topology optimization. *Journal of Computational Science and Technology*, 2(1):46–55, 2008.

[108] Liu, K., Tovar, A., Nutwell, E., and Detwiler, D. Thin-walled compliant mechanism component design assisted by machine learning and multiple surrogates. In *SAE 2015 World Congress & Exhibition*, 2015.

[109] Livermore Software Technology Corporation (LSTC). LS-DYNA theory manual. P. O. Box 712 Livermore, California 94551-0712, 2015.

[110] Lu, J. and Chen, Y. Performance evaluation of particle swarm optimization and solid isotropic material with penalization in topology optimization. In *Computational Intelligence for Measurement Systems and Applications (CIMSA), 2012 IEEE International Conference on*, pages 97–101. IEEE, 2012.

[111] Luh, G.-C. and Chueh, C.-H. Multi-modal topological optimization of structure using immune algorithm. *Computer Methods in Applied Mechanics and Engineering*, 193(36):4035–4055, 2004.

[112] Luh, G.-C., Lin, C.-Y., and Lin, Y.-S. A binary particle swarm optimization for continuum structural topology optimization. *Applied Soft Computing*, 11(2):2833 – 2844, 2011.

[113] Madeira, J., Rodrigues, H., and Pina, H. Multiobjective topology optimization of structures using genetic algorithms with chromosome repairing. *Structural and Multidisciplinary Optimization*, 32(1):31–39, 2006.

[114] Madeira, J. A., Rodrigues, H., and Pina, H. Multi-objective optimization of structures topology by genetic algorithms. *Advances in Engineering Software*, 36(1):21–28, 2005.

[115] Madeira, J., Pina, H., and Rodrigues, H. GA topology optimization using random keys for tree encoding of structures. *Structural and Multidisciplinary Optimization*, 40(1-6):227–240, 2010.

[116] Mattheck, C. *Design in Nature Learning from Trees*. Springer-Verlag Berlin Heidelberg, 1998.

[117] Mayer, R. R., Kikuchi, N., and Scott, R. A. Application of topological optimization techniques to structural crashworthiness. *International Journal for Numerical Methods in Engineering*, 39(8): 1383–1403, 1996.

[118] Michell, A. G. M. LVIII. the limits of economy of material in frame-structures. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 8(47):589–597, 1904.

[119] Miguel Nunes, R. K., Marcelo Kobayashi. On a cellular division method for layout optimization and sub-system placement. In *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2012.

[120] Mlejnek, H. Some aspects of the genesis of structures. *Structural Optimization*, 5(1-2):64–69, 1992.

[121] Mozumder, C. and Renaud, J. E. Cost and mass optimization for crashworthiness design of shell-based structure using hybrid cellular automata. In *Proceedings of the 50tAIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Palm Spring, CA*, 2009.

[122] Mozumder, C. K. *Topometry optimization of sheet metal structures for crashworthiness design using hybrid cellular automata*. PhD thesis, University of Notre Dame, 2010.

[123] Myers, R. H., Montgomery, D. C., and Anderson-Cook, C. M. *Response surface methodology: process and product optimization using designed experiments*, volume 705. John Wiley & Sons, 2009.

[124] Neri, F. and Cotta, C. Memetic algorithms and memetic computing optimization: A literature review. *Swarm and Evolutionary Computation*, 2(0):1 – 14, 2012.

[125] Norato, J., Bendsøe, M., Haber, R., and Tortorelli, D. A topological derivative method for topology optimization. *Structural and Multidisciplinary Optimization*, 33(4-5):375–386, 2007.

[126] Olschinka, C. and Schumacher, A. Graph based topology optimization of crashworthiness structures. *PAMM*, 8(1):10029–10032, 2008.

[127] Ortmann, C. *Entwicklung eines graphen- und heuristikbasierten Verfahrens zur Topologieoptimierung von Profilquerschnitten für Crashlastfälle*. PhD thesis, Bergische Universität Wuppertal, 2015.

[128] Ortmann, C. and Schumacher, A. Graph and heuristic based topology optimization of crash loaded structures. *Structural and Multidisciplinary Optimization*, 47(6):839–854, 2013.

[129] Padhye, N. Topology optimization of compliant mechanism using multi-objective particle swarm optimization. In *Proceedings of the 2008 GECCO conference companion on Genetic and evolutionary computation*, GECCO '08, pages 1831–1834, New York, NY, USA, 2008. ACM.

[130] Park, G.-J. Technical overview of the equivalent static loads method for non-linear static response structural optimization. *Structural and Multidisciplinary Optimization*, 43:319–337, 2011. 10.1007/s00158-010-0530-x.

[131] Patel, N. M. *Crashworthiness design using topology optimization*. PhD thesis, University of Notre Dame, 2007.

[132] Patel, N. M., Kang, B.-S., Renaud, J. E., and Tovar, A. Crashworthiness design using topology optimization. *Journal of Mechanical Design*, 131:061013, 2009.

[133] Patnaik, S. N. and Hopkins, D. A. Optimality of a fully stressed design. *Computer Methods in Applied Mechanics and Engineering*, 165(1):215–221, 1998.

[134] Pedersen, C. B. W. Topology optimization design of crushed 2d-frames for desired energy absorption history. *Structural and Multidisciplinary Optimization*, 25(5-6):368–382, 2003.

[135] Pedersen, C. B. W. Topology optimization for crashworthiness of frame structures. *International Journal of Crashworthiness*, 8(1): 29–39, 2003.

[136] Pedersen, C. B. W. Topology optimization of 2d-frame structures with path-dependent response. *International Journal for Numerical Methods in Engineering*, 57(10):1471–1501, 2003.

[137] Pedersen, C. B. W. Crashworthiness design of transient frame structures using topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 193(6-8):653 – 678, 2004.

[138] Pedro, H.-T. C. and Kobayashi, M. H. On a cellular division method for topology optimization. *International Journal for Numerical Methods in Engineering*, 88(11):1175–1197, 2011.

[139] Rechenberg, I. Cybernetic solution path of an experimental problem. Technical report, Royal Air Force Establishment, 1965.

[140] Rechenberg, I. *Evolutionsstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.* Fommann-Holzboog, 1973.

[141] References to CMA-ES Applications. `https://www.lri.fr/~hansen/cmaapplications.pdf`. Accessed at: March 10, 2016.

[142] Reisinger, J., Bahçeci, E., Karpov, I., and Miikkulainen, R. Coevolving strategies for general game playing. In *CIG*, pages 320–327. IEEE, 2007.

[143] Rios, L. M. and Sahinidis, N. V. Derivative-free optimization: A review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56(3):1247–1293, 2013.

[144] Roux, W. *LS-TaSC^{TM} Topology and Shape Computation for LS-DYNA User's Manual.* Livermore Software Technology Corporation, November 2011.

[145] Rozvany, G. I. A critical review of established methods of structural topology optimization. *Structural and Multidisciplinary Optimization*, 37(3):217–237, 2009.

[146] Rudolph, G. Evolutionary strategies. In Rozenberg, G., Bäck, T., and Kok, J. N., editors, *Handbook of Natural Computing*, pages 673–698. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[147] Sandgren, E., Jensen, E., and Welton, J. Topological design of structural components using genetic optimization methods. In *Winter Annual Meeting of the American Society of Mechanical Engineers*, volume AMD-115 of *Sensitivy Analysis and Optimization with Numerical Methods*, pages 31–43, 1990.

[148] Sauter, M., Kress, G., Giger, M., and Ermanni, P. Complex-shaped beam element and graph-based optimization of compliant mechanisms. *Structural and Multidisciplinary Optimization*, 36(4):429–442, 2008.

[149] Sauter, M. *A Graph-based Optimization Method for the Design of Compliant Mechanisms and Structures.* PhD thesis, Swiss Federal Institute of Technology Zurich, 2008.

[150] Saxena, A. On optimization of 2d compliant mechanisms using honeycomb discretization with material-mask overlay strategy. In *ASME 2007 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 1331–1341. American Society of Mechanical Engineers, 2007.

[151] Saxena, A. A material-mask overlay strategy for continuum topology optimization of compliant mechanisms using honeycomb discretization. *J. Mech. Des*, 130:082304, 2008.

[152] Saxena, A. An adaptive material mask overlay method: Modifications and investigations on binary, well connected robust compliant continua. *J. Mech. Des*, 133(4):041004, 2011.

[153] Saxena, A. Are circular shaped masks adequate in adaptive mask overlay topology synthesis method? *Journal of Mechanical Design*, 133(1):011001, 2011.

[154] Saxena, A. Topology design with negative masks using gradient search. *Structural and Multidisciplinary Optimization*, 44(5):629–649, 2011.

[155] Schoenauer, M. Shape representations and evolution schemes. *Evolutionary Programming*, 5, 1996.

[156] Schumacher, A. *Optimierung mechanischer Strukturen.* Springer Verlag Berlin Heidelberg, 2005.

[157] Schumacher, A. Parameter-based topology optimization for crashworthiness structures. In *CDROM Proc. of the World Congress of Structural and Multidisciplinary Optimization (WCSMO-6), Rio-de-Janeiro, Brazil.* Citeseer, 2005.

[158] Schwefel, H.-P. *Kybernetische Evolution als Strategie der experimentellen Forschung in der Strömungstechnik.* Dipl.-ing. thesis, Technical University of Berlin, HermannFöttinger–Institute for Fluid Dynamics, 1965.

[159] Schwefel, H.-P. P. *Evolution and optimum seeking: the sixth generation.* John Wiley & Sons, Inc., 1993.

[160] Sharma, D. and Deb, K. Generation of compliant mechanisms using hybrid genetic algorithm. *Journal of The Institution of Engineers (India): Series C*, 95(4):295–307, 2014.

[161] Sharma, D., Deb, K., and Kishore, N. An improved initial population strategy for compliant mechanism designs using evolutionary optimization. In *ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 1175–1184. American Society of Mechanical Engineers, 2008.

[162] Sharma, D., Deb, K., and Kishore, N. Towards generating diverse topologies of path tracing compliant mechanisms using a local search based multi-objective genetic algorithm procedure. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 2004–2011. IEEE, 2008.

[163] Sharma, D., Deb, K., and Kishore, N. Domain-specific initial population strategy for compliant mechanisms using customized genetic algorithm. *Structural and Multidisciplinary Optimization*, 43(4):541–554, 2011.

[164] Shim, P. Y. and Manoochehri, S. Generating optimal configurations in structural design using simulated annealing. *International journal for numerical methods in engineering*, 40(6):1053–1069, 1997.

[165] Shinde, S., Detwiler, D., and Tovar, A. Structural optimization of thin-walled tubular structures using weighted multi-objective approach. In *10th World Congress on Structural and Multidisciplinary Optimization*, 2013.

[166] Sigmund, O. *Design of material structures using topology optimization*. PhD thesis, Department of Solid Mechanics, Technical University of Denmark, 1994.

[167] Sigmund, O. and Petersson, J. Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Struct. Optim.*, 16:68–75, 1998.

[168] Sigmund, O. A 99 line topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization*, 21:120–127, 2001.

[169] Sigmund, O. Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization*, 33(4-5):401–424, 2007.

[170] Sigmund, O. On the usefulness of non-gradient approaches in topology optimization. *Structural and Multidisciplinary Optimization*, 43: 589–596, 2011.

[171] Sigmund, O. and Maute, K. Topology optimization approaches. *Structural and Multidisciplinary Optimization*, 48(6):1031–1055, 2013.

[172] Smalikho, O. and Olhofer, M. Co-evolution of sensory system and signal processing for optimal wing shape control. In *Applications of Evolutionary Computation*, pages 805–816. Springer, 2014.

[173] Smola, A. J. and Schölkopf, B. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[174] Soto, C. A. Structural topology optimisation: from minimising compliance to maximising energy absorption. *International Journal of Vehicle Design*, 25(1-2):142–163, 2001.

[175] Spall, J. C. *Introduction to stochastic search and optimization: estimation, simulation, and control*, volume 65. John Wiley & Sons, Inc., 2003.

[176] Stanford, B., Beran, P., and Kobayashi, M. Aeroelastic optimization of flapping wing venation: A cellular division approach. *AIAA Journal*, 50(4):938–951, 2012.

[177] Stanford, B., Beran, P., , and Kobayashi, M. Simultaneous topology optimization of membrane wings and their compliant flapping mechanisms. *AIAA Journal*, 51(6):1431–1441, 2013.

[178] Stanley, K. O. and Miikkulainen, R. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2): 99–127, 2002.

[179] Stanley, K. O. Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines*, 8(2):131–162, 2007.

[180] Stanley, K. O., Cornelius, R., and Miikkulainen, R. Real-time learning in the NERO video game. In Young, R. M. and Laird, J. E., editors, *AIIDE*, pages 159–160. AAAI Press, 2005.

[181] Steiner, T. *Artificial evolutionary development*. PhD thesis, Universität Bielefeld, 2010.

[182] Steiner, T., Jin, Y., and Sendhoff, B. A cellular model for the evolutionary development of lightweight material with an inner structure. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2008)*, New York, 2008. ACM.

[183] Steiner, T., Trommler, J., Brenn, M., Jin, Y., and Sendhoff, B. Global shape with morphogen gradients and motile polarized cells. In *Proceedings of the 2009 IEEE Congress on Evolutionary Computation, Trondheim*. IEEE, 2009.

[184] Svanberg, K. The method of moving asymptotes-a new method for structural optimization. *International journal for numerical methods in engineering*, 24(2):359–373, 1987.

[185] Svanberg, K. A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM journal on optimization*, 12(2):555—573, 2002.

[186] Tai, K. and Akhtar, S. Structural topology optimization using a genetic algorithm with a morphological geometric representation scheme. *Structural and Multidisciplinary Optimization*, 30(2):113–127, 2005.

[187] Tai, K. and Chee, T. Design of structures and compliant mechanisms by evolutionary optimization of morphological representations of topology. *Journal of Mechanical Design*, 122(4):560–566, 2000.

[188] Tai, K., Cui, G. Y., and Ray, T. Design synthesis of path generating compliant mechanisms by evolutionary optimization of topology and shape. *Journal of Mechanical Design*, 124(3):492–500, 2002.

[189] The MathWorks, Inc. MATLAB release 2014b. Natick, Massachusetts, United States, 2014.

[190] Tovar, A. *Bone remodeling as a hybrid cellular automaton optimization process.* PhD thesis, University of Notre Dame, 2004.

[191] Tovar, A., Patel, N. M., Niebur, G. L., Sen, M., and Renaud, J. E. Topology optimization using a hybrid cellular automaton method with local control rules. *Journal of Mechanical Design*, 128:1205–1216, 2006.

[192] Valsalam, V., Hiller, J., MacCurdy, R., Lipson, H., and Miikkulainen, R. Constructing controllers for physical multilegged robots using the ENSO neuroevolution approach. *Evolutionary Intelligence*, 5(1):45–56, 2012.

[193] van Dijk, N., Maute, K., Langelaar, M., and van Keulen, F. Level-set methods for structural topology optimization: a review. *Structural and Multidisciplinary Optimization*, 48(3):437–472, 2013.

[194] Vapnik, V. *The Nature of Statistical Learning Theory.* Springer, 1995.

[195] Volz, K. *Physikalisch begründete Ersatzmodelle für die Crashoptimierung von Karosseriestrukturen in frühen Projektphasen.* PhD thesis, Technische Universität München, 2011.

[196] Wang, K., Shengyin; Tai. A constraint handling strategy for bit-array representation GA in structural topology optimization. *High Performance Computation for Engineered Systems (HPCES)*, 2004.

[197] Wang, M. Y., Wang, X., and Guo, D. A level set method for structural topology optimization. *Computer methods in applied mechanics and engineering*, 192(1):227–246, 2003.

[198] Wang, N. and Tai, K. Target matching problems and an adaptive constraint strategy for multiobjective design optimization using genetic algorithms. *Computers & Structures*, 88(19-20):1064 – 1076, 2010.

[199] Wang, N. and Zhang, X. Topology optimization of compliant mechanisms using pairs of curves. *Engineering Optimization*, 47(11):1497–1522, 2015.

[200] Wang, N., Yang, Y., and Tai, K. Optimization of structures under load uncertainties based on hybrid genetic algorithm. In *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 4039–4044, June 2008.

[201] Wang, N. F., Zhang, X., and Yang, Y. A hybrid genetic algorithm for constrained multi-objective optimization under uncertainty and target matching problems. *Applied Soft Computing*, 13(8):3636–3645, 2013.

[202] Wang, S. Y., Tai, K., and Quek, S. T. Topology optimization of piezoelectric sensors/actuators for torsional vibration control of composite plates. *Smart Materials and Structures*, 15(2):253, 2006.

[203] Wang, S. Y., Tai, K., and Wang, M. Y. An enhanced genetic algorithm for structural topology optimization. *International Journal for Numerical Methods in Engineering*, 65(1):18–44, 2006.

[204] Wang, S. and Tai, K. Graph representation for structural topology optimization using genetic algorithms. *Computers & Structures*, 82 (20-21):1609 – 1622, 2004.

[205] Wang, S. and Tai, K. Structural topology design optimization using genetic algorithms with a bit-array representation. *Computer Methods in Applied Mechanics and Engineering*, 194(36-38):3749 – 3770, 2005.

[206] Witowski, K., Erhart, A., Schumacher, P., and Müllerschön, H. Topology optimization for crash. In *12th International LS-Dyna Users Conference*, 2012.

[207] Witowski, K., Müllerschön, H., Erhart, A., Schumacher, P., and Anakiev, K. Topology and topometry optimization of crash applications with the equivalent static load method. In *13th International LS-Dyna Users Conference*, 2014.

[208] Wolpert, D. H. and Macready, W. G. No free lunch theorems for search. Technical report, Technical Report SFI-TR-95-02-010, Santa Fe Institute, 1995.

[209] Woon, S. Y., Tong, L., Querin, O. M., and Steven, G. P. Effective optimisation of continuum topologies through a multi-ga system. *Computer Methods in Applied Mechanics and Engineering*, 194(30-33):3416–3437, 2005.

[210] Wu, C. Y. and Ku, C. C. Stress-enhanced clonal selection algorithm for structural topology optimization. *International Journal for Numerical Methods in Engineering*, 89(8):957–974, 2012.

[211] Wu, C.-Y. and Tseng, K.-Y. Topology optimization of structures using modified binary differential evolution. *Structural and Multidisciplinary Optimization*, 42(6):939–953, 2010.

[212] Xie, Y. M. and Steven, G. P. *Evolutionary Structural Optimization*. Springer, 1997.

[213] Xie, Y. M. and Steven, G. P. A simple evolutionary procedure for structural optimization. *Computers & structures*, 49(5):885–896, 1993.

[214] Yang, R. Multidiscipline topology optimization. *Computers & Structures*, 63(6):1205–1212, 1997.

[215] Yao, X. Evolving artificial neural networks. *Proceedings of the IEEE*, 87:1423–1447, 1999.

[216] Yi, S.-I., Lee, H.-A., and Park, G.-J. Optimization of a structure with contact conditions using equivalent loads. *Journal of mechanical science and technology*, 25(3):773–782, 2011.

[217] Zhou, M. and Rozvany, G. The COC algorithm, part II: topological, geometrical and generalized shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 89(1):309–336, 1991.

Curriculum Vitae

Personal Details

Name	Nikola Aulig
Date of birth	Feb 06, 1984
Plance of birth	Karlsruhe

Scientific Development

since 2017	Senior Scientist at Honda Research Institute Europe GmbH, Offenbach/Main, Germany
2011-2016	Scientist at Honda Research Institute Europe GmbH, Offenbach/Main, Germany
2014-2017	Doctoral candidate at Technische Universtität Darmstadt, Germany, Control Methods and Robotics, in cooperation with Honda Research Institute Europe GmbH
2011	Diploma in Electrical Engineering and Information Technology at Technische Universität Darmstadt, Germany
2003	Abitur, Max-Planck-Gymnasium, Karlsruhe, Germany